Disjoint NP-Pairs and Propositional Proof Systems

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum naturalium
(Dr. rer. nat.)
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät II
Humboldt-Universität zu Berlin

von
Herr Dipl.-Math. Olaf Beyersdorff
geboren am 18.08.1973 in Greifswald

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Christoph Markschies

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät II:
Prof. Dr. Wolfgang Coy

Gutachter:

1. Prof. Dr. Johannes Köbler
2. Prof. Dr. Martin Grohe
3. Prof. Dr. Pavel Pudlák

Tag der mündlichen Prüfung: 12. Juli 2006

Abstract

Disjoint NP-pairs are an interesting complexity theoretic concept with important applications in cryptography and propositional proof complexity. In this dissertation we explore the connection between disjoint NP-pairs and propositional proof complexity. This connection is fruitful for both fields. Various disjoint NP-pairs have been associated with propositional proof systems which characterize important properties of these systems, yielding applications to areas such as automated theorem proving. Further, conditional and unconditional lower bounds for the separation of disjoint NP-pairs can be translated to results on lower bounds to the length of propositional proofs. In this way disjoint NP-pairs have substantially contributed to the understanding of propositional proof systems.

Conversely, this dissertation aims to transfer proof-theoretic knowledge to the theory of NP-pairs to gain a more detailed understanding of the structure of the class of disjoint NP-pairs and in particular of the NP-pairs defined from propositional proof systems. For a proof system P we introduce the complexity class $\mathsf{DNPP}(P)$ of all disjoint NP-pairs for which the disjointness of the pair is efficiently provable in the proof system P. We exhibit structural properties of proof systems which make the previously defined canonical NP-pairs of these proof systems hard or complete for $\mathsf{DNPP}(P)$. Moreover we demonstrate that non-equivalent proof systems can have equivalent canonical pairs and that depending on the properties of the proof systems different scenarios for $\mathsf{DNPP}(P)$ and the reductions between the canonical pairs exist. As an important tool for our investigation we use the connection of propositional proof systems and disjoint NP-pairs to theories of bounded arithmetic.

We also investigate a natural generalization of disjoint NP-pairs: instead of pairs we consider k-tuples of pairwise disjoint NP-sets. In our main result in this part we show that complete disjoint NP-pairs exist if and only if complete disjoint k-tuples of NP-sets exist for all $k \geq 2$. Further, this is equivalent to the existence of a propositional proof system in which the disjointness of all tuples is shortly provable. We also show that a strengthening of this conditions characterizes the existence of optimal proof systems.

Keywords:
disjoint NP-pairs, propositional proof systems, bounded arithmetic, complexity theory

Zusammenfassung

Die Theorie disjunkter NP-Paare, die auf natürliche Weise statt einzelner Sprachen Paare von NP-Mengen zum Objekt ihres Studiums macht, ist vor allem wegen ihrer Anwendungen in der Kryptografie und Beweistheorie interessant. Im Zentrum dieser Dissertation steht die Analyse der Beziehung zwischen disjunkten NP-Paaren und aussagenlogischen Beweissystemen. Haben die Anwendungen der NP-Paare in der Beweistheorie maßgeblich das Verständnis aussagenlogischer Beweissysteme gefördert, so beschreiten wir in dieser Arbeit gewissermaßen den umgekehrten Weg, indem wir Methoden der Beweistheorie zur genaueren Untersuchung des Verbands disjunkter NP-Paare heranziehen.

Insbesondere ordnen wir jedem Beweissystem P eine Klasse $\mathsf{DNPP}(P)$ von NP-Paaren zu, deren Disjunktheit in dem Beweissystem P mit polynomiell langen Beweisen gezeigt werden kann. Zu diesen Klassen $\mathsf{DNPP}(P)$ zeigen wir eine Reihe von Resultaten, die illustrieren, dass robust definierten Beweissystemen sinnvolle Komplexitätsklassen $\mathsf{DNPP}(P)$ entsprechen. Als wichtiges Hilfsmittel zur Untersuchung aussagenlogischer Beweissysteme und der daraus abgeleiteten Klassen von NP-Paaren benutzen wir die Korrespondenz starker Beweissysteme zu erststufigen arithmetischen Theorien, die gemeinhin unter dem Schlagwort beschränkte Arithmetik zusammengefasst werden.

In der Praxis trifft man statt auf zwei häufig auf eine größere Zahl konkurrierender Bedingungen. Daher widmen wir uns der Erweiterung der Theorie disjunkter NP-Paare auf disjunkte Tupel von NP-Mengen. Unser Hauptergebnis in diesem Bereich besteht in der Charakterisierung der Fragen nach der Existenz optimaler Beweissysteme und vollständiger NP-Paare mit Hilfe disjunkter Tupel.

Schlagwörter:
disjunkte NP-Paare, aussagenlogische Beweissysteme, beschränkte Arithmetik, Komplexitätstheorie

Contents

Acknowlegdements

First of all I am very grateful to my advisor Johannes Köbler for his constant support and advice during the preparation of this dissertation. He continously guided my work with detailed suggestions and helpful comments.

For encouraging support I am also very grateful to Jan Krajíček and Pavel Pudlák from the Mathematical Institute of the Academy of Sciences in Prague. Many helpful and stimulating conversations during several visits to Prague and during five Fall Schools at Pec pod Sněžkou have considerably deepened my understanding of propositional proof complexity. It was also Pavel Pudlák from whom I first learned about the beautiful theory of disjoint NP-pairs at the Fall School 2001.

Chapter 1

Introduction

In den Werken des Menschen, wie in denen der Natur, sind eigentlich die Absichten vorzüglich der Aufmerksamkeit wert.

Johann Wolfgang Goethe

Disjoint NP-pairs are an interesting complexity theoretic concept with important applications in cryptography and propositional proof complexity. Even though the foundations of the theory of disjoint NP-pairs were already laid in the 80's it was only during recent years that disjoint NP-pairs have fully come into the focus of complexity theoretic research.

In this dissertation we explore the connection between disjoint NP-pairs and propositional proof complexity. This connection is fruitful for both fields. Various disjoint NP-pairs have been associated with propositional proof systems which characterize important properties of these systems, yielding applications to areas such as automated theorem proving. Further, conditional and unconditional lower bounds for the separation of disjoint NP-pairs can be translated to results on lower bounds to the length of propositional proofs. In this way disjoint NP-pairs have substantially contributed to the understanding of propositional proof systems.

Conversely, this dissertation aims to transfer proof-theoretic knowledge to the theory of NP-pairs to gain a more detailed understanding of the structure of the class of disjoint NP-pairs and in particular of the NP-pairs defined from propositional proof systems. Let us formulate the fruitfulness of this approach as the main thesis of this dissertation:

Disjoint NP-pairs are intimately connected to propositional proof systems. Although the definition of disjoint NP-pairs is completely complexity theoretic

with no reference to proof systems the theory of disjoint NP*-pairs is best analysed and explained by logical methods.*

To substantiate this claim we will try to provide an overall picture of the theory of disjoint NP-pairs, including also a presentation of results in our framework which have been previously obtained by different techniques.

But before we start to explain this material in more detail let us make some remarks on the development of the subject.

1.1 Computational Complexity, Bounded Arithmetic, and Propositional Logic

Using logical methods has a rich tradition in complexity theory. In particular there are very close relations between computational complexity, propositional proof complexity and bounded arithmetic, and the central tasks in these areas of separating complexity classes, proving lower bounds to the length of propositional proofs and separating arithmetic theories can be understood as different approaches towards the same problem. Let us dwell on these relations a little as methods from all three areas will be used in this dissertation.

Computational complexity studies the amount of resources which is required for the solution of computational tasks. A major open problem in the field is the precise comparison between deterministic and nondeterministic computations, leading for polynomial time computations to the famous P/NP-problem formulated already more than 30 years ago by Cook (Coo71) and Karp (Kar72). The solution of the P/NP-problem has far reaching implications, mainly because, starting with Cook's completeness result, a vast number of problems with immense practical relevance have been shown to be NP-complete. Despite enormous efforts the separation of complexity classes remains elusive today. Current techniques such as diagonalization and circuit lower bounds are all ineffectual, with even theoretical evidence supporting the failure of these approaches (BGS75; RR94).

A different, logic oriented way of studying complexity classes is through weak fragments of arithmetic, usually referred to as theories of bounded arithmetic. These fragments have the right strength to formalize and reason about efficient computations. More formally, definable functions and predicates in these theories can be used to characterize functions and languages from standard complexity classes, the most prominent example being the hierarchy of theories S_2^i and T_2^i defined by Buss (Bus86) which correspond to the computational strength of the levels of the polynomial hierarchy. These

strong relations between the theories S_2^i and PH were established by a series of witnessing theorems due to Buss (Bus86; Bus90) and Krajíček, Pudlák and Takeuti (KPT91). In particular Krajíček, Pudlák and Takeuti proved that a collapse of the hierarchy of the theories S_2^i implies a collapse of PH. Later Buss (Bus95) and Zambella (Zam96) independently strengthened this result by showing that $S_2 = \bigcup_{i=1}^{\infty} S_2^i$ is finitely axiomatizable if and only if PH collapses and this collapse is provable in S_2.

Bounded arithmetic is also closely connected to propositional proof systems. This connection was first developed by Cook (Coo75) who gave a translation of bounded first order formulas into polynomial size sequences of propositional formulas. Different and refined translations have later been introduced by Paris and Wilkie (PW85) as well as by Krajíček and Pudlák (KP90). These translations allow the use of logical and in particular model theoretic machinery to obtain lower bounds to the size of propositional proofs, which constitutes the main objective in propositional proof complexity. In particular Ajtai (Ajt94) successfully used these methods to show superpolynomial lower bounds to the proof size in bounded-depths Frege systems (cf. Theorem 3.5.4 for the general framework). Together with later improvements this currently forms one of the strongest results about propositional proof systems. Another connection to bounded arithmetic comes from the reflection principles which are arithmetic formulas stating the consistency of propositional proof systems. On the one hand these formulas are candidates for the separation of arithmetic theories, on the other hand proving reflection principles in arithmetic theories yields simulations between propositional proof systems. This technique was first used by Krajíček and Pudlák (KP89) to show the equivalence of extended Frege and substitution Frege systems.

The circle back to computational complexity is completed with the results of Cook and Reckhow (CR79), who show that polynomially bounded proof systems exist if and only if NP is closed under complementation. Thus, similarly as the circuit complexity approach, proving lower bounds to successively stronger systems can be understood as a way to address the P/NP-question by non-uniform methods. In fact, the relationship between proof complexity and computational complexity extends to other complexity classes than NP. Köbler, Messner and Torán (KMT03) have shown that the problem on the existence of complete sets for promise classes like $\mathsf{NP} \cap \mathsf{coNP}$ or BPP can be reformulated as questions about proof systems.

1.2 Disjoint NP-Pairs

Disjoint NP-pairs, which are the central topic of this dissertation, enjoy connections to all three fields mentioned in the last paragraph. Like many complexity theoretic notions the idea to study disjoint pairs of languages instead of single objects originates in recursion theory. In the 80's Joachim Grollmann and Alan Selman (GS88) introduced disjoint NP-pairs as a complexity theoretic concept in connection to questions concerning the foundations of cryptography. Grollmann and Selman developed many of the central notions including reductions and separators for pairs.

The connection of disjoint NP-pairs to propositional proof systems was first made by Alexander Razborov (Raz94) who associated a canonical disjoint NP-pair with a proof system. Razborov used the correspondence to bounded arithmetic for his investigation of disjoint NP-pairs, namely he studied classes of NP-pairs which are provably disjoint in some arithmetic theory. He gave a list of theories and corresponding proof systems for which his canonical pairs are complete for the respective class of disjoint NP-pairs. In particular this included the Frege and extended Frege system. Razborov also raised the question whether there exists a complete disjoint NP-pair. Similarly as for other promise classes there are currently no completeness results for the class of all disjoint NP-pairs. Unfortunately Razborov's work remained as a technical report and therefore did not receive wider attention.

The next step was taken by Pavel Pudlák. In his very influential paper (Pud03) Pudlák demonstrated that disjoint NP-pairs can characterize different properties of propositional proof systems. In particular Pudlák showed that Razborov's canonical pairs are tightly linked to the automatizability of the proof system, a concept that is of great relevance for automated theorem proving. Pudlák also characterizes the feasible interpolation property by a disjoint NP-pair. Feasible interpolation, introduced by Jan Krajíček (Kra97), provides a general method for proving lower bounds to the proof size in weak proof systems. In fact, proving these lower bounds rests again on lower bounds to the monotone circuit complexity required for the separation of disjoint NP-pairs as provided by Razborov (Raz85) and Alon and Boppana (AB87). Pudlák further shows that also weak systems like resolution give rise to interesting canonical pairs with robust properties.

These applications attracted further complexity theoretic research on the structure of the class of disjoint NP-pairs. Most notably, Glaßer, Selman, Sengupta and Zhang investigated the structure of disjoint NP-pairs by complexity theoretic techniques in a series of papers (GSSZ04; GSS05; GSZ05). In particular they worked on the problem on the existence of complete disjoint NP-pairs. Glaßer et al. (GSS05) gave a characterization in terms of

uniform enumerations of disjoint NP-pairs and also proved that the answer to the problem does not depend on the reductions used, i.e. there are reductions for pairs which vary in strength but are equivalent with respect to the existence of complete pairs. Köbler, Messner and Torán (KMT03) had already previously linked this problem with the existence of complete sets for other promise classes, showing in particular that the existence of optimal proof systems implies the existence of complete disjoint NP-pairs under strong reductions. However, Glaßer et al. (GSSZ04) construct an oracle relative to which there exist complete pairs but optimal proof systems do not exist. Hence, the problems on the existence of optimal proof systems and of complete disjoint NP-pairs appear to be of different strength.

In this dissertation we continue the line of research of Razborov (Raz94) and Pudlák (Pud03) which focuses on the connection between disjoint NP-pairs and propositional proof systems. While we hope to have contributed to the understanding of disjoint NP-pairs we feel that the subject as a whole is still in an early stage of its development and is considerably less understood than other complexity theoretic concepts. However, we think that it is especially the interdisciplinary nature of the field, allowing the use of completely different techniques from complexity theory and from both propositional and first-order logic, that together with diverse applications will stimulate future research on this fascinating subject.

1.3 Organization of the Dissertation and Obtained Results

In this section we will provide an overview of this dissertation.

We start in Chap. 2 by recalling some background information about propositional proof systems. This includes the definition of those proof systems that will play a major role in further chapters: resolution and extensions of Frege systems. In Sect. 2.6 we define and investigate natural properties of proof systems which we use throughout this dissertation. These properties are of logical nature: it should be feasible to carry out basic operations like modus ponens and substitutions in the proof system. Most of these properties have probably been used before in several contexts. For a subject like disjoint NP-pairs which can be seen both from a proof complexity and also from a computational complexity perspective we feel that it is important to be precise about the exact conditions that are imposed on proof systems. If complexity theorists state a theorem like

For all propositional proof systems the following holds . . . ,

then they really mean that this theorem holds for all functions computed by deterministic polynomial time Turing machines which have as their range the set of tautologies. If on the other hand people from proof complexity use this phrase it is often implicitly understood from the context that the result only holds for some class of meaningful proof systems, operating for example with formulas and enjoying some basic closure properties. Therefore, combining results from both worlds without being conscious about the context in which they are applicable may result in confusion (at least this happened to me once). We therefore try to be rather pedantic in always listing explicitly all assumptions that are made on the proof system. Actually, the results of this dissertation support the view that the Cook-Reckhow frame work for propositional proof systems in its full generality is too broad for the study of naturally defined classes of disjoint NP-pairs. It therefore seems to be natural to concentrate on proof systems on which further conditions are imposed.

In Chap. 3 we explain the correspondence between bounded arithmetic and propositional proof systems. We do not give all details but instead concentrate on those issues that we need for Chap. 4 to explore the structure of disjoint NP-pairs. In the first four sections of Chap. 3 we outline the formalization of syntactic concepts such as propositional formulas and propositional proof systems in arithmetic theories. We also describe in detail the translation of first-order formulas into sequences of propositional formulas as given by Cook (Coo75) and by Krajíček and Pudlák (KP90). We then proceed in Sect. 3.5 with the general correspondence between arithmetic theories and propositional proof systems as defined by Krajíček and Pudlák (KP90). In Sect. 3.6 we explain this correspondence for the theory S_2^1 and the extended Frege system as well as for extensions of EF by additional axioms. Sect. 3.7 is again devoted to the general correspondence from (KP90). We give a refined analysis of proof systems admitting a corresponding arithmetic theory. We call such proof systems regular and exhibit sufficient conditions for the regularity of propositional proof systems. These results are particularly useful for our investigations into disjoint NP-pairs in the following chapter.

The material from Chap. 3 and most of the results proven there are certainly known to the experts in the field. To my knowledge there is, however, no account that develops the general correspondence between bounded arithmetic and propositional proof systems in full detail as we need it for subsequent chapters. The original source (KP90) introduces this correspondence in a very condensed way, and it is unfortunately left out from the standard reference (Kra95). There is, however, a number of beautiful introductory expositions, most notably (Pud98) and (Kra01b).

Chapter 4 on disjoint NP-pairs comprises the main part of this dissertation. We start with the relevant definitions and make some observations

about the simulation order of disjoint NP-pairs. Section 4.4 then explains in detail how NP-pairs can be used to characterize properties of propositional proof systems. The converse approach to exploit proof-theoretic machinery for the analysis of disjoint NP-pairs starts with Sect. 4.5. We investigate a slight modification of the first-order arithmetic representations of disjoint NP-pairs defined by Razborov (Raz94). We also define more general propositional representations for NP-pairs and associate with any propositional proof system P a subclass $\mathsf{DNPP}(P)$ of NP-pairs for which the disjointness is provable with short P-proofs. Somewhat surprisingly, under suitable conditions on P these non-uniform classes $\mathsf{DNPP}(P)$ equal their uniform versions which are defined via arithmetic representations.

In Sect. 4.6 we investigate the class $\mathsf{DNPP}(P)$, showing that under reasonable assumptions on the proof system P this class is closed under reductions for pairs and possesses hard or complete pairs in form of Razborov's canonical pair, Pudlák's interpolation pair and other pairs associated with the proof system. The properties of the classes $\mathsf{DNPP}(P)$ are decisively influenced by the closure properties of the underlying proof system. We demonstrate that proof systems P with different properties give rise to different scenarios for $\mathsf{DNPP}(P)$ and the reductions between the NP-pairs associated with P.

We proceed with the connection between the simulation order of propositional proof systems and disjoint NP-pairs. As all information about the proof lengths is coded in the canonical pair the simulations between proof systems are reflected in reductions between NP-pairs and specifically between canonical pairs. Among other things this implies that the existence of optimal proof systems implies the existence of complete NP-pairs. On the other hand this connection is not as tight as one might hope for. In Sect. 4.13 we provide different ways to construct non-equivalent proof systems with equivalent canonical pairs. A first example for this situation is due to Pudlák (Pud03). Here we search for general conditions on proof systems that yield a collapse between their canonical pairs. In particular we analyse a weak notion of simulation for proof systems introduced in (KP89) but not much studied elsewhere. This simulation is provably weaker than the ordinary reduction between proof systems but is equivalent with respect to the existence of optimal proof systems. We show that all proof systems that are equivalent with respect to this weak simulation possess equivalent canonical pairs.

Chapter 5 mentions two applications of the theory of disjoint NP-pairs. The first application dates back to Grollmann and Selman (GS88) and connects disjoint NP-pairs and public-key crypto systems. The second application relates to a recent program for the search of hard tautologies that are obtained from pseudorandom generators. Proving lower bounds to the proof size of strong proof systems like Frege systems and their extensions

is a major challenge in propositional proof complexity. Even to exhibit viable candidates for formulas without polynomial size proofs in Frege systems seems to be a complicated task (BBP95; Pud91). Krajíček (Kra01a; Kra01b) and independently Alekhnovich, Ben-Sasson, Razborov and Wigderson (ABSRW04) suggested to employ pseudorandom generators as the basis for hard tautologies. So far this program has proved to be successful for weak systems like resolution (ABSRW04; Kra04). In Sect. 5.2 we give a characterization of the hardness of these formulas for strong proof systems in terms of disjoint NP-pairs. Whether such a characterization helps to solve the original problem remains open. But it provides further evidence that disjoint NP-pairs are applicable to interesting, seemingly unconnected areas.

In the last chapter we investigate a natural generalization of disjoint NP-pairs: instead of pairs we consider k-tuples of pairwise disjoint NP-sets. Concepts such as reductions and separators are smoothly generalized from pairs to k-tuples. Our main interest in this chapter is the characterization of the two problems on the existence of optimal proof systems and complete NP-pairs in terms of disjoint k-tuples of NP-sets. In particular we address the question whether there exist complete disjoint k-tuples under different reductions. Considering this problem it is easy to see that the existence of complete k-tuples implies the existence of complete l-tuples for $l \leq k$: the first l components of a complete k-tuple are complete for all l-tuples. Conversely, it is a priori not clear how to construct a complete k-tuple from a complete l-tuple for $l < k$. Therefore it might be tempting to conjecture that the existence of complete k-tuples forms a hierarchy of assumptions of increasing strength for greater k. However, we show that this does not happen: there exist complete disjoint NP-pairs if and only if there exist complete disjoint k-tuples of NP-sets for all $k \geq 2$, and this is even true under reductions of different strength. Further, we prove that this is equivalent to the existence of a propositional proof system in which the disjointness of all k-tuples with respect to suitable propositional representations of these tuples is provable with short proofs. We also characterize the existence of optimal proof systems with a similar but apparently stronger condition.

We achieve this by extending the connection between proof systems and NP-pairs to k-tuples. We define propositional representations for k-tuples and introduce the complexity classes $\mathsf{DNPP}_k(P)$ of all disjoint k-tuples of NP-sets that are representable in the system P. We show that these classes are closed under our reductions for k-tuples. Further, we define k-tuples from propositional proof systems which serve as hard languages for $\mathsf{DNPP}_k(P)$. In particular we generalize the interpolation pair from (Pud03) and demonstrate that even these generalized variants still capture the feasible interpolation property of the proof system.

1.4 Published Parts

Most of the results on disjoint NP-pairs from Chap. 4 have appeared in the conference publications (Bey04a) (Foundations of Software Technology and Theoretical Computer Science, FSTTCS) and (Bey06a) (Theory and Applications of Models of Computation, TAMC). The article (Bey06a) also contains some results from Sects. 3.7 and 3.8 on the general correspondence between arithmetic theories and propositional proof systems. A shortened version of Chap. 6 about disjoint tuples of NP-sets is published as the conference paper (Bey06b) (International Computer Science Symposium in Russia, CSR).

Full versions of these conference contributions have appeared as technical reports at the Electronic Colloquium on Computational Complexity (ECCC) (Bey04b; Bey05a; Bey05b). The report (Bey04b) also contains the characterization of the hardness of the τ-formulas in terms of NP-pairs as explained in Sect. 5.2.

Chapter 2

Propositional Proof Systems

Alles Gescheite ist schon gedacht worden, man muß nur versuchen, es noch einmal zu denken.

Johann Wolfgang Goethe

This chapter is largely of preliminary nature. We review relevant concepts from propositional logic and proof complexity. The emphasis is laid on propositional proof systems and their properties.

We refrain from defining the complexity theoretic notation as we follow the general conventions. For background information on notions from computational complexity we refer to (BDG88) and (Pap94).

2.1 Propositional Logic

In this section we will review some notions from propositional logic with the purpose to fix the notation. The language of propositional logic consists of a set of propositional variables

$$\text{Var} = \{p_1, p_2, p_3 \ldots\} \ ,$$

the connectives $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$, the constants $0, 1$ and the brackets $(,)$. The set of *propositional formulas* Form is inductively defined as follows:

1. Every variable $p \in \text{Var}$ and constant $0, 1$ is in Form.
2. If $\varphi, \psi \in \text{Form}$, then also $\neg\varphi, (\varphi \vee \psi), (\varphi \wedge \psi), (\varphi \rightarrow \psi), (\varphi \leftrightarrow \psi) \in \text{Form}$.

We follow the usual conventions to omit $(,)$ from formulas, i.e. $\wedge$ binds stronger than $\vee$ which is stronger than $\rightarrow$ and $\leftrightarrow$. For multiple connectives of the same type brackets are associated from right to left.

Formulas can be coded in binary and we denote by $|\varphi|$ the length of the encoding of φ.

For a propositional formula φ we define $\mathrm{Var}(\varphi)$ as the set of propositional variables occurring in φ. A *propositional assignment* α is a mapping

$$\alpha : \mathrm{Var} \rightarrow \{0,1\} \ .$$

An assignment α can be extended to a mapping

$$\alpha' : \mathrm{Form} \rightarrow \{0,1\}$$

via:

1. $\alpha'(p) = \alpha(p)$ for $p \in \mathrm{Var}$.
2. $\alpha'(0) = 0$ and $\alpha'(1) = 1$.
3. $\alpha'(\neg\varphi) = 1 - \alpha'(\varphi)$ for $\varphi \in \mathrm{Form}$.
4. $\alpha'(\varphi \wedge \psi) = \begin{cases} 1 & \text{if } \alpha'(\varphi) = \alpha'(\psi) = 1 \\ 0 & \text{otherwise.} \end{cases}$
   for $\varphi, \psi \in \mathrm{Form}$ and similarly for the other connectives.

We call α an *assignment for a formula* φ if α is a mapping

$$\alpha : \mathrm{Var}(\varphi) \rightarrow \{0,1\}$$

which can be extended to an ordinary assignment by defining α arbitrarily on $\mathrm{Var} \setminus \mathrm{Var}(\varphi)$.

We say that α is a *satisfying assignment* for a formula φ if $\alpha'(\varphi) = 1$. We denote this by $\alpha \models \varphi$. The set of all satisfiable formulas is denoted by

$$\mathrm{SAT} = \{\varphi \in \mathrm{Form} \mid \text{ there exists an assignment } \alpha \text{ such that } \alpha \models \varphi\} \ .$$

A formula φ is a *tautology* if it is satisfied by all assignments, denoted by $\models \varphi$. The set of all tautologies is

$$\mathrm{TAUT} = \{\varphi \in \mathrm{Form} \mid \ \models \alpha\} \ .$$

It is a classical result of Cook (Coo71) that SAT is NP-complete, while TAUT is complete for coNP.

Instead of the constants 0 and 1 we also use the symbols $\bot$ and $\top$ to denote a fixed unsatisfiable formula and a fixed tautology, respectively.

If $\Phi \subseteq \text{Form}$ and $\varphi \in \text{Form}$, then we write $\Phi \models \varphi$ if all assignments that satisfy all formulas from Φ also satisfy φ.

A *substitution* σ is a mapping

$$\sigma : \text{Var} \to \text{Form} \ .$$

If a substitution σ only substitutes variables by constants, i.e.

$$\sigma(p) \in \{p, 0, 1\} \quad \text{for all } p \in \text{Var},$$

then we call σ a *substitution by constants*.

A substitution σ can be extended to a mapping

$$\sigma' : \text{Form} \to \text{Form}$$

defined by:

1. $\sigma'(p) = \sigma(p)$ for $p \in \text{Var}$.
2. $\sigma'(0) = 0$ and $\sigma'(1) = 1$.
3. $\sigma'(\neg\varphi) = \neg\sigma'(\varphi)$ for $\varphi \in \text{Form}$.
4. $\sigma'(\varphi \wedge \psi) = \sigma'(\varphi) \wedge \sigma'(\psi)$ for $\varphi, \psi \in \text{Form}$ and similarly for the other connectives.

To simplify the notation we will identify σ and σ' in the following.

2.2 Propositional Proof Complexity

Propositional proof systems were defined in a very general way by Cook and Reckhow in (CR79) as polynomial time functions P which have as its range the set of all tautologies.

Definition 2.2.1 (Cook, Reckhow (CR79)) *A* propositional proof system *is a polynomial time computable function P with* $\text{rng}(P) = \text{TAUT}$.

A string π with $P(\pi) = \varphi$ is called a P-proof of the tautology φ. The intuition behind this definition is that given a proof it should be easy to determine which formula is actually proven and to verify the correctness of the proof. Nevertheless it might be difficult to generate proofs for a given

formula and proofs might be very long compared to the size of the formula proven.

Probably the simplest proof system is the *truth-table system* that proves formulas by checking all propositional assignments. In the sense of Definition 2.2.1 proofs in the truth-table system consist of the proven formula φ together with a string $1^{2^{|\mathrm{Var}(\varphi)|}}$. As most formulas require exactly exponential proof size in this system it is neither very interesting from the application oriented nor from the proof complexity perspective.

But also all the usually studied proof systems are captured by the above definition. Let us illustrate this by an example. One of the most widely used proof systems is the *resolution calculus* and its variants introduced by Davis and Putnam (DP60) and Robinson (Rob65). Resolution is a refutation system that operates with clauses which are finite sets of negated or unnegated variables called literals. A clause is associated with the disjunction of the literals it contains and a set of clauses is associated with the conjunction of its clauses. Therefore finite sets of clauses correspond to propositional formulas in conjunctive normal form.

A clause is satisfied by a propositional assignment if at least one literal of the clause is satisfied by the assignment. Therefore by definition the empty clause is unsatisfiable. A resolution proof shows the unsatisfiability of a set of clauses by starting with these clauses and deriving new clauses by the resolution rule

$$\frac{C \cup \{p\} \quad D \cup \{\neg p\}}{C \cup D}$$

until the empty clause is derived.

At first glance the resolution systems does not seem to fit into the Cook-Reckhow framework of propositional proof systems because it is a refutation system and can furthermore only refute formulas in CNF. But we can associate with resolution the following function Res:

$$Res(\pi) = \begin{cases} \varphi & \text{if } \pi = (\varphi, C_1, \ldots, C_k) \text{ where } \varphi \text{ is a formula in DNF} \\ & \text{and } C_1, \ldots C_k \text{ is a resolution refutation of the set} \\ & \text{of clauses for } \neg\varphi \\ \varphi & \text{if } \pi = (\varphi, 1^m) \text{ with } m \geq 2^{|\varphi|} \text{ and } \varphi \in \mathrm{TAUT} \\ \top & \text{otherwise.} \end{cases}$$

The second line of the definition is needed to prove formulas which are not in disjunctive normal form whereas the last line is incorporated because by definition every string π has to be interpreted as a proof of some formula. Res is computable in polynomial time because in line 2 of its definition the parameter m is big enough to allow testing $\varphi \in \mathrm{TAUT}$ by checking all

assignments. Hence *Res* is a proof system in accordance with the above general definition.

Another common way to extend the resolution system from a proof system for formulas in DNF to a proof system for all propositional tautologies is to transfer the formula to an equivalent formula in DNF, either by direct translation or by using new auxiliary variables (cf. (Bus98b) for the details).

By the notation

$$P \vdash_{\leq m} \varphi$$

we indicate that there is a P-proof of φ of length $\leq m$. If Φ is a set of propositional formulas we write

$$P \vdash_* \Phi$$

if there is a polynomial p such that $P \vdash_{\leq p(|\varphi|)} \varphi$ for all $\varphi \in \Phi$. If $\Phi = \{\varphi_n | n \geq 0\}$ is a sequence of formulas we also write $P \vdash_* \varphi_n$ instead of $P \vdash_* \Phi$.

Proof systems can be compared according to their strength by the notion of simulation. Given two proof systems P and S we say that S *simulates* P (denoted by $P \leq S$) if there exists a polynomial p such that for all tautologies φ and P-proofs π of φ there is a S-proof π' of φ with $|\pi'| \leq p(|\pi|)$ (KP89). If such a proof π' can even be computed from π in polynomial time we say that S *p-simulates* P and denote this by $P \leq_p S$ (CR79). If $P \leq S$, then we will often simply say that S is stronger than P. As usual we say that P and S are equivalent (denoted by $P \equiv S$) if $P \leq S$ and $S \leq P$. The relation $\equiv_p$ is defined similarly. It is clear that $\equiv$ and $\equiv_p$ are equivalence relations on the set of all proof systems. Their equivalence classes are called *degrees*.

A proof system is called *(p-)optimal* if it (p-)simulates all proof systems. Whether or not optimal proof systems exist is an open problem posed by Krajíček and Pudlák (KP89). But it is known that $\mathsf{NE} = \mathsf{coNE}$ is a sufficient condition for the existence of optimal proof systems (KP89). On the other hand Köbler, Messner and Torán (KMT03) showed that optimal proof systems imply complete sets for various promise classes like $\mathsf{NP} \cap \mathsf{coNP}$. This may be interpreted as evidence that optimal systems do not exist.

A proof system P is called *polynomially bounded* if there is a polynomial p such that $P \vdash_{\leq p(|\varphi|)} \varphi$ for all tautologies φ. Given the general notion of a proof system from Definition 2.2.1 a proof system is simply a nondeterministic procedure that accepts TAUT. Hence polynomially bounded proof systems correspond to NP-algorithms for TAUT. This connection to complexity theory is made precise by the following theorem of Cook and Reckhow from their seminal paper (CR79).

Theorem 2.2.2 (Cook, Reckhow (CR79)) *There exists a polynomially bounded proof system if and only if* $\mathsf{NP} = \mathsf{coNP}$.

Proof. For the first direction let P be a polynomially bounded proof system with bounding polynomial p. Consider the following algorithm:

```
1  Input:  a formula φ
2  guess  π ∈ Σ^{≤p(|φ|)}
3  IF  P(π) = φ  THEN accept ELSE reject
```

Obviously the above algorithm is a nondeterministic polynomial time algorithm for TAUT. Because TAUT is coNP-complete this implies NP = coNP.

For the other direction assume that NP = coNP. Hence there exists a nondeterministic polynomial time Turing machine M that accepts TAUT. Let the polynomial p bound the running time of M. Then

$$P(\pi) = \begin{cases} \varphi & \text{if } \pi \text{ codes an accepting computation of } M(\varphi) \\ \top & \text{otherwise} \end{cases}$$

is a proof system which is polynomially bounded by p. □

From this theorem the following approach which is sometimes referred to as the Cook-Reckhow program is derived. To separate NP from coNP (and hence also P from NP) it is sufficient to establish for stronger and stronger proof systems that they are not polynomially bounded. Although it is debatable whether this approach is indeed a sensible strategy to show NP ≠ coNP the above theorem is often used as a complexity theoretic justification for the interest in lower bounds to the lengths of proofs for a diversity of proof systems.

Figure 2.1 depicts some of the most common proof systems together with their simulation relations. A line between proof systems indicates that the lower proof system is simulated by the higher system in Fig. 2.1. Moreover all the proof systems below the dashed line have also been separated, i.e. the simulations do not hold in the opposite direction. The dashed line shows the current frontier in the search for super-polynomial lower bounds to the proof length, i.e. for all systems below the line sequences of formulas are known that do not admit polynomial size proofs in the respective proof systems, whereas for the systems above the line there is currently no information about non-trivial lower bounds to the proof size available. A detailed description of the proof systems depicted in Fig. 2.1 together with information on lower bounds can be found in the surveys (Pud98) and (Urq95).

2.3 Frege Systems and Their Extensions

In this section we will describe Frege systems and their extensions. These are strong proof systems that will play a central role for the rest of this work.

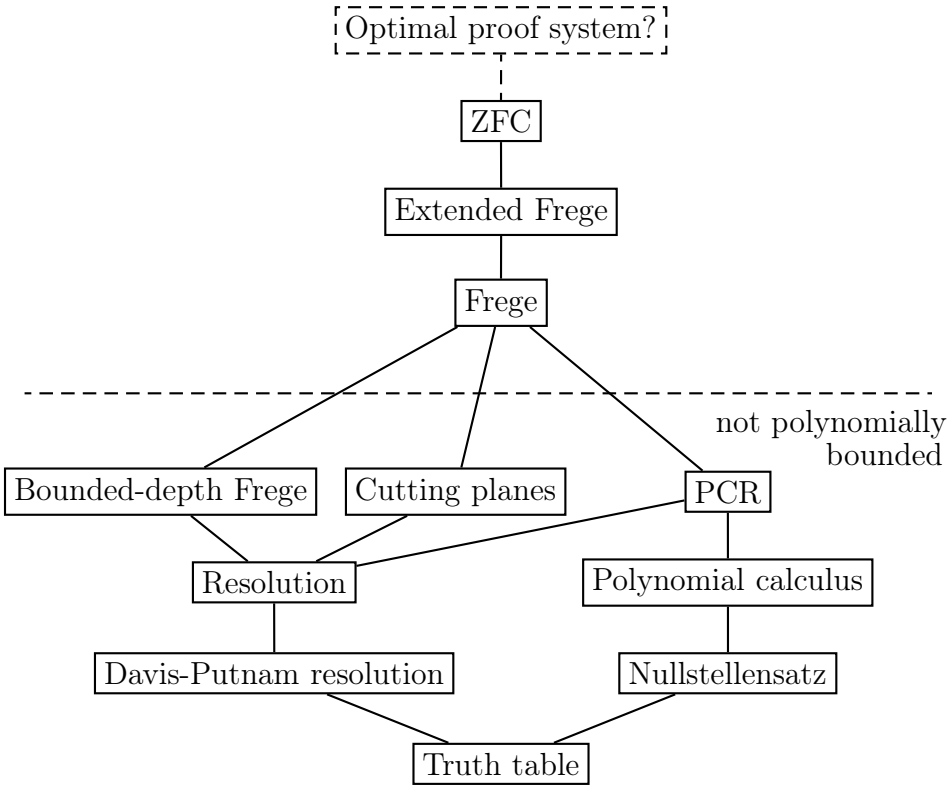

Figure 2.1: The simulation order of propositional proof systems

Unfortunately our present knowledge about proof complexity questions of these systems is still very poor.

Frege systems derive formulas using axioms and rules. In texts on classical logic these systems are usually referred to as Hilbert-style systems but in propositional proof complexity it has become customary to call them Frege systems (CR79).

A *Frege rule* is a $(k+1)$-tuple $(\varphi_0, \varphi_1 \ldots, \varphi_k)$ of propositional formulas such that

$$\{\varphi_1, \varphi_2, \ldots, \varphi_k\} \models \varphi_0 \ .$$

The standard notation for rules is

$$\frac{\varphi_1 \quad \varphi_2 \quad \cdots \quad \varphi_k}{\varphi_0} \ .$$

A Frege rule with $k = 0$ is called a *Frege axiom*.

A formula ψ_0 can be derived from formulas $\psi_1, \ldots, \psi_k$ by a Frege rule $(\varphi_0, \varphi_1 \ldots, \varphi_k)$ if there exists a substitution σ such that

$$\sigma(\varphi_i) = \psi_i \quad \text{for } i = 0, \ldots, k \ .$$

Let $\mathcal{F}$ be a finite set of Frege rules. An *$\mathcal{F}$-proof* of a formula φ from a set of propositional formulas Φ is a sequence $\varphi_1, \ldots, \varphi_l = \varphi$ of propositional formulas such that for all $i = 1, \ldots, l$ one of the following holds:

1. $\varphi_i \in \Phi$ or
2. there exist numbers $1 \leq i_1 \leq \ldots \leq i_k < i$ such that φ_i can be derived from $\varphi_{i_1}, \ldots, \varphi_{i_k}$ by a Frege rule from $\mathcal{F}$.

We denote this by $\mathcal{F} : \Phi \vdash \varphi$.

$\mathcal{F}$ is called *complete* if for all formulas φ

$$\models \varphi \quad \Longleftrightarrow \quad \mathcal{F} : \emptyset \vdash \varphi \ .$$

$\mathcal{F}$ is called *implicationally complete* if for all $\varphi \in \text{Form}$ and $\Phi \subseteq \text{Form}$

$$\Phi \models \varphi \quad \Longleftrightarrow \quad \mathcal{F} : \Phi \vdash \varphi \ .$$

$\mathcal{F}$ is a *Frege system* if $\mathcal{F}$ is implicationally complete.

Without poof we note that the following set of axioms which we have taken from (Bus98b)

$$
\begin{array}{l}
p_1 \to (p_2 \to p_1) \\
(p_1 \to p_2) \to (p_1 \to (p_2 \to p_3)) \to (p_1 \to p_3) \\
p_1 \to p_1 \vee p_2 \\
p_2 \to p_1 \vee p_2 \\
(p_1 \to p_3) \to (p_2 \to p_3) \to (p_1 \vee p_2 \to p_3) \\
(p_1 \to p_2) \to (p_1 \to \neg p_2) \to \neg p_1 \\
\neg\neg p_1 \to p_1 \\
p_1 \wedge p_2 \to p_1 \\
p_1 \wedge p_2 \to p_2 \\
p_1 \to p_2 \to p_1 \wedge p_2 \\
(p_1 \leftrightarrow p_2) \to (p_1 \to p_2) \\
(p_1 \leftrightarrow p_2) \to (p_2 \to p_1) \\
(p_1 \to p_2) \to (p_2 \to p_1) \to (p_1 \leftrightarrow p_2) \\
1 \leftrightarrow p_1 \vee \neg p_1 \\
0 \leftrightarrow \neg 1
\end{array}
$$

together with the modus ponens rule

$$\frac{\varphi \quad \varphi \to \psi}{\psi}$$

is an example for a Frege system.

This definition leaves much freedom to design individual Frege systems but if we are only interested in the lengths of proofs there is only one Frege system as already noted by Cook and Reckhow (CR79).

Theorem 2.3.1 (Cook, Reckhow (CR79)) *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be Frege systems. Then $\mathcal{F}_1 \equiv_p \mathcal{F}_2$.*

Proof. It is enough to show $\mathcal{F}_1 \leq_p \mathcal{F}_2$. Let $\mathcal{F}_1 = \{R_1, \dots R_n\}$ with the rules R_i

$$\frac{\varphi_1^i \quad \dots \quad \varphi_{k_i}^i}{\varphi_0^i} \ .$$

Because of the correctness of the rules R_i and the implicational completeness of $\mathcal{F}_2$ there exist $\mathcal{F}_2$-proofs π_i of φ_0^i from $\{\varphi_1^i, \dots \varphi_{k_i}^i\}$.

Let π be an $\mathcal{F}_1$-proof of the formula φ and let

$$\frac{\psi_1 \quad \dots \quad \psi_{k_i}}{\psi_0}$$

be an application of the rule R_i in π via the substitution σ, i.e. $\sigma(\varphi_j^i) = \psi_j$ for $j = 0, \ldots, k_i$. Applying σ to each formula in the proof π_i gives an $\mathcal{F}_2$-proof of ψ_0 from $\{\psi_1, \ldots, \psi_{k_i}\}$. Performing this transformation for every application of an $\mathcal{F}_1$-rule in π we efficiently construct an $\mathcal{F}_2$-proof of φ which is only polynomially longer than π. □

Now we describe the extensions of Frege systems as introduced in (CR79). Let $\mathcal{F}$ be a Frege system. An *extended Frege proof* of φ from $\Phi \subseteq \mathrm{Form}$ is a sequence $(\varphi_1, \ldots, \varphi_l = \varphi)$ of propositional formulas such that for each $i = 1, \ldots, l$ one of the following holds:

1. $\varphi_i \in \Phi$ or
2. φ_i has been derived by an $\mathcal{F}$-rule or
3. $\varphi_i = q \leftrightarrow \psi$ where ψ is an arbitrary propositional formula and q is a new propositional variable that does not occur in φ, ψ and φ_j for $1 \leq j < i$.

The introduction of the extension rule 3 allows the abbreviation of possibly complex formulas by variables. Hence using this rule for formulas which appear very often in an $\mathcal{F}$-proof can substantially reduce the proof size.

Analogously as in Theorem 2.3.1 it follows that all extended Frege systems are polynomially equivalent. Therefore we will henceforth only speak of the *extended Frege system* and denote it by EF.

It is clear that EF simulates Frege systems but whether EF is indeed a strictly stronger system is an open problem.

Another way to enhance the power of Frege systems is to allow substitutions not only for axioms but also for all formulas that have been derived in Frege proofs. This is accomplished by introducing the *substitution rule*

$$\frac{\varphi}{\sigma(\varphi)}$$

which allows to derive $\sigma(\varphi)$ for an arbitrary substitution σ from the earlier proven formula φ. Augmenting Frege systems by this substitution rule we arrive at the *substitution Frege system* SF.

SF is polynomially equivalent to EF. While $EF \leq_p SF$ is relatively easy to see (CR79) the transformation of SF-proofs to EF-proofs on the propositional level is quite involved (KP89). But using the correspondence to bounded arithmetic this simulation can be shown very elegantly (Dow85; KP89). We will discuss this in more detail in Sect. 3.

As mentioned earlier, with present knowledge we cannot exclude the possibility that EF or even Frege systems are optimal. Still it is interesting to

look for ways to further strengthen the power of EF. This can be done by adding further axioms to EF. Since we already know that all formulations of Frege and extended Frege systems are polynomially equivalent adding any finite number of new axioms cannot produce stronger systems. Therefore we have to add infinitely many new axioms to the system. In order to define in this way a correct proof system in the sense of Definition 2.2.1 we have to require that this infinite set of axioms can be checked in polynomial time. We will explain this in a more general context.

We call a proof system *line based* if proofs in the system consist of sequences of formulas, and formulas in such a sequence are derived from earlier formulas in the sequence by the rules available in the proof system. Most of the studied proof systems like resolution, cutting planes and Frege systems are line based in this sense.

In the following we will often enhance line based proof systems by additional axioms. We will do this in two different ways. Let Φ be a set of tautologies which can be decided in polynomial time. By $P + \Phi$ we denote the proof system P augmented by the possibility to use all formulas from Φ as axiom schemes. This means that formulas from Φ as well as substitution instances of these formulas can be freely introduced as new lines in $P + \Phi$ -proofs. In contrast to this standard notation we denote by $P \cup \Phi$ the proof system that extends P by formulas from Φ as new axioms. The difference to $P + \Phi$ is that in $P \cup \Phi$ we are only allowed to use formulas from Φ but not their substitution instances in proofs.

2.4 Efficient Deduction

The deduction theorem of propositional logic states that in a Frege system $\mathcal{F}$ a formula ψ is provable from a formula φ if and only if $\varphi \to \psi$ is provable in $\mathcal{F}$. Because proof complexity is focusing on the length of proofs it is interesting to analyse how the proof length is changing in the deduction theorem. An $\mathcal{F}$-proof of $\varphi \to \psi$ together with the axiom φ immediately yields the formula ψ with one application of modus ponens. Therefore it is only interesting to ask for the increase in proof length when constructing a proof of $\varphi \to \psi$ from an $\mathcal{F}$-proof of ψ with the extra axiom φ. This was analysed in detail in (Bon93; BB93).

Since the deduction property makes sense for all line based proof systems we give the following general definition.

Definition 2.4.1 *A line based proof system P allows* efficient deduction *if*

there exists a polynomial p such that for all finite sets of tautologies Φ

$$P \cup \Phi \vdash_{\leq m} \psi \quad \textit{implies} \quad P \vdash_{\leq p(m+m')} (\bigwedge_{\varphi\in\Phi} \varphi) \to \psi$$

where $m' = |\bigwedge_{\varphi\in\Phi} \varphi|$.

Along the lines of the proof of the deduction theorem for Frege systems (see e.g. (Kra95)) we can prove:

Theorem 2.4.2 (Deduction theorem for EF) *The extended Frege system EF allows efficient deduction. Moreover, given an $EF\cup\Phi$-proof of a formula ψ for finite $\Phi \subseteq$ TAUT we can construct an EF-proof of $(\bigwedge_{\varphi\in\Phi} \varphi) \to \psi$ in polynomial time.*

Proof. For every $\mathcal{F}$-rule

$$R_i = \frac{\psi_1 \quad \dots \quad \psi_r}{\psi}$$

in EF we fix an $\mathcal{F}$-proof π_i of the tautology

$$((q \to \psi_1) \wedge \ldots \wedge (q \to \psi_r)) \to (q \to \psi) \ .$$

Note that for $r = 0$ this also includes the case that R_i is an axiom scheme.

Let $\varphi_1, \ldots, \varphi_n$ be tautologies and let $(\theta_1, \ldots, \theta_k)$ be a proof of ψ of size m in the system $EF \cup \{\varphi_1, \ldots, \varphi_n\}$. Let $m' = \sum_{i=1}^{n} |\varphi_i|$. By induction on j we construct proofs of the implications

$$(\bigwedge_{i=1}^{n} \varphi_i) \to \theta_j \ .$$

We distinguish three cases on how the formula θ_j was derived.

If θ_j was inferred from $\theta_{j_1}, \ldots, \theta_{j_r}$ by the $\mathcal{F}$-rule R_i, then we can get from π_i an $\mathcal{F}$-proof of size $O(m' + |\theta_j| + \sum_{l=1}^{r} |\theta_{j_l}|)$ of the tautology

$$(((\bigwedge_{i=1}^{n} \varphi_i) \to \theta_{j_1}) \wedge \ldots \wedge ((\bigwedge_{i=1}^{n} \varphi_i) \to \theta_{j_r})) \to ((\bigwedge_{i=1}^{n} \varphi_i) \to \theta_j) \ .$$

Combining all the earlier proved implications

$$(\bigwedge_{i=1}^{n} \varphi_i) \to \theta_{j_l},\ l = 1, \ldots, r$$

by conjunctions and using modus ponens we get the desired implication

$$(\bigwedge_{i=1}^{n} \varphi_i) \to \theta_j$$

in a proof of size $O(m + m')$.

If θ_j is one of the formulas from $\{\varphi_1, \ldots, \varphi_n\}$, then we get $(\bigwedge_{i=1}^{n} \varphi_i) \to \theta_j$ in a proof of size $O(m')$.

Let now θ_j be derived by the extension rule, i.e.

$$\theta_j = (q \leftrightarrow \theta)$$

with a new variable q. In this case we also use the extension rule to get $(q \leftrightarrow \theta)$ and then derive

$$(\bigvee_{i=1}^{n} \neg\varphi_i) \vee (q \leftrightarrow \theta) = (\bigwedge_{i=1}^{n} \varphi_i) \to (q \leftrightarrow \theta) \ .$$

in a proof of size $O(m' + |\theta|)$. □

2.5 The Propositional Sequent Calculus

Historically one of the first and best analysed proof systems is Gentzen's sequent calculus (Gen35). The sequent calculus is widely used both for propositional and first-order logic. Here we will describe the propositional sequent calculus LK. The basic objects of the sequent calculus are *sequents*

$$\varphi_1, \ldots, \varphi_m \longrightarrow \psi_1, \ldots, \psi_k \ .$$

Formally these are ordered pairs of two sequences of propositional formulas separated by the symbol $\longrightarrow$. The sequence $\varphi_1, \ldots, \varphi_m$ is called the *antecedent* and $\psi_1, \ldots, \psi_k$ is called the *succedent*. These cedents are usually denoted by letters like Γ and Δ. An assignment α satisfies a sequent

$$\Gamma \longrightarrow \Delta$$

if

$$\alpha \models \bigvee_{\varphi \in \Gamma} \neg\varphi \vee \bigvee_{\psi \in \Delta} \psi \ .$$

The sequence $\emptyset \longrightarrow \Delta$ having empty antecedent is abbreviated as $\longrightarrow \Delta$. Likewise $\Gamma \longrightarrow$ abbreviates $\Gamma \longrightarrow \emptyset$. Sequences of the form

$$A \longrightarrow A, \quad 0 \longrightarrow, \quad \longrightarrow 1$$

are called *initial sequents*. The sequent calculus LK uses the following set of rules:

1. weakening rules

$$\frac{\Gamma \longrightarrow \Delta}{A, \Gamma \longrightarrow \Delta} \quad \text{and} \quad \frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, A}$$

2. exchange rules

$$\frac{\Gamma_1, A, B, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, B, A, \Gamma_2 \longrightarrow \Delta} \quad \text{and} \quad \frac{\Gamma \longrightarrow \Delta_1, A, B, \Delta_2}{\Gamma \longrightarrow \Delta_1, B, A, \Delta_2}$$

3. contraction rules

$$\frac{\Gamma_1, A, A, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, A, \Gamma_2 \longrightarrow \Delta} \quad \text{and} \quad \frac{\Gamma \longrightarrow \Delta_1, A, A, \Delta_2}{\Gamma \longrightarrow \Delta_1, A, \Delta_2}$$

4. $\neg$: introduction rules

$$\frac{\Gamma \longrightarrow \Delta, A}{\neg A, \Gamma \longrightarrow \Delta} \quad \text{and} \quad \frac{A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \neg A}$$

5. $\wedge$: introduction rules

$$\frac{A, \Gamma \longrightarrow \Delta}{A \wedge B, \Gamma \longrightarrow \Delta} \quad \text{and} \quad \frac{A, \Gamma \longrightarrow \Delta}{B \wedge A, \Gamma \longrightarrow \Delta}$$

$$\text{and} \quad \frac{\Gamma \longrightarrow \Delta, A \qquad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \wedge B}$$

6. $\vee$: introduction rules

$$\frac{A, \Gamma \longrightarrow \Delta \qquad B, \Gamma \longrightarrow \Delta}{A \vee B, \Gamma \longrightarrow \Delta}$$

$$\text{and} \quad \frac{\Gamma \longrightarrow \Delta, A}{\Gamma \longrightarrow \Delta, A \vee B} \quad \text{and} \quad \frac{\Gamma \longrightarrow \Delta, A}{\Gamma \longrightarrow \Delta, B \vee A}$$

7. cut-rule

$$\frac{\Gamma \longrightarrow \Delta, A \qquad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$$

Similarly as in Frege systems an *LK-proof* of a propositional formula φ is a derivation of the sequent

$$\longrightarrow \varphi$$

from initial sequents by the above rules. Without proof we note that the above set of rules specifies a proof system that is complete for the set of all tautologies not containing the connectives $\rightarrow$ and $\leftrightarrow$ (see (Kra95)).

As Frege systems can be easily transformed into the sequent formulation a straightforward analysis shows that Frege systems and the Gentzen calculus LK suitably extended for formulas containing $\rightarrow, \leftrightarrow$ polynomially simulate each other.

Proposition 2.5.1 (Cook, Reckhow (CR79)) *The propositional sequent calculus LK and Frege systems are polynomially equivalent.*

2.6 Natural Properties of Proof Systems

Although we are interested in information on general proof systems we will very often in the course of this dissertation consider proof systems satisfying some additional properties. The conditions are of logical nature: it should be feasible to carry out basic operations like modus ponens or substitutions by constants in the proof system. These are very natural requirements that are met by most of the studied proof systems. Nevertheless the general definition of propositional proof systems above permits a great variety of proof systems that violate these conditions.

Definition 2.6.1 *A proof system P is* closed under modus ponens *if there exists a polynomial p such that for all formulas φ and ψ*

$$P \vdash_{\leq m} \varphi \quad \textit{and} \quad P \vdash_{\leq n} \varphi \to \psi \quad \textit{imply} \quad P \vdash_{\leq p(m+n)} \psi \ .$$

This definition is a weak form of saying that modus ponens is available as a rule in the proof system. If P is closed under modus ponens, then we can apply modus ponens constantly many times with only polynomial increase in the proof length. In Frege systems, however, modus ponens can be used arbitrarily often whereas with our definition this might produce exponentially long proofs. Therefore a stronger form of closure under modus ponens is given in the following definition:

Definition 2.6.2 *A proof system P is* closed under multiple applications of modus ponens *if there exists a constant c such that for all formulas φ and ψ*

$$P \vdash_{\leq m} \varphi \quad \textit{and} \quad P \vdash_{\leq n} \varphi \to \psi \quad \textit{imply} \quad P \vdash_{\leq m+n+|\psi|+c} \psi \ .$$

The application we have in mind for this definition is the following. Suppose we have $P \vdash_{\leq n_i} \varphi_i$ for $i = 1, \ldots, k$ and also

$$P \ \vdash_{\leq m} \ \varphi_1 \to \varphi_2 \to \ldots \to \varphi_{k+1} \ .$$

If P is closed under multiple applications of modus ponens, then we get a P-proof of φ_{k+1} of size $\leq m + kc + \sum_{i=1}^{k} n_i + |\varphi_{i+1}|$ which is polynomial in n_i, m and k. Using closure under modus ponens in the form of Definition 2.6.1 we would only get an exponential upper bound on the proof size of φ_{k+1}.

We could have also defined closure under multiple applications of modus ponens in a slightly weaker fashion: if $P \vdash_{\leq n} \varphi_i$ for $i = 1, \ldots, k$ and $P \vdash_{\leq n} \varphi_1 \to \varphi_2 \to \ldots \to \varphi_{k+1}$, then we get $P \vdash_{\leq p(n)} \varphi_{k+1}$ for some fixed polynomial p. Definition 2.6.2 implies this condition but is apparently a stricter formulation which exactly resembles the situation in Frege systems. What is actually the right formulation of such closure properties might also depend on the particular application.

If π is a Frege proof of a formula φ, then we can prove substitution instances $\sigma(\varphi)$ of φ by applying the substitution σ to every formula in the proof π. This leads us to the general concept of closure of a proof system under substitutions.

Definition 2.6.3 *P is* closed under substitutions *if there exists a polynomial q such that*

$$P \vdash_{\leq n} \varphi \quad \textit{implies} \quad P \vdash_{\leq q(n+|\sigma(\varphi)|)} \sigma(\varphi)$$

for all formulas φ and all substitutions σ.

Likewise we say that P is closed under substitutions by constants *if there exists a polynomial q such that*

$$P \vdash_{\leq n} \varphi(\bar{x}, \bar{y}) \quad \textit{implies} \quad P \vdash_{\leq q(n)} \varphi(\bar{a}, \bar{y})$$

for all formulas $\varphi(\bar{x}, \bar{y})$ and constants $\bar{a} \in \{0,1\}^{|\bar{x}|}$.

Modus ponens and substitutions are transformations on proofs which we can also define in a more constructive fashion. As we will need these versions at some places we make the following definition.

Definition 2.6.4 *A proof system P is* efficiently closed under modus ponens *if there exists a polynomial time computable algorithm that takes as input P-proofs π_1, π_2 of formulas φ and $\varphi \to \psi$ and outputs a P-proof π_3 of ψ. If in addition we always have $|\pi_3| \leq |\pi_1| + |\pi_2| + |\psi| + c$ for some fixed constant c, then we say that the system P is* efficiently closed under multiple applications of modus ponens.

Similarly, we say that P is efficiently closed under substitutions *if we can transform any P-proof of a formula φ in polynomial time to a P-proof of $\sigma(\varphi)$ for arbitrary substitutions σ.*

Occasionally we will also consider other properties. We say that a proof system *evaluates formulas without variables* if formulas using only constants but no propositional variables have polynomially long proofs. As this is true even for truth-table evaluations all proof systems simulating the truth-table

system evaluate formulas without variables. A system P is *closed under disjunctions* if there is a polynomial q such that

$$P \vdash_{\leq m} \varphi \quad \text{implies} \quad P \vdash_{\leq q(m+|\psi|)} \varphi \vee \psi \quad \text{and} \quad P \vdash_{\leq q(m+|\psi|)} \psi \vee \varphi$$

for arbitrary formulas ψ. Similarly we say that a proof system P is *closed under conjunctions* if there is a polynomial q such that

$$P \vdash_{\leq m} \varphi \wedge \psi \quad \text{implies} \quad P \vdash_{\leq q(m)} \varphi \quad \text{and} \quad P \vdash_{\leq q(m)} \psi \ ,$$

and

$$P \vdash_{\leq m} \varphi \quad \text{and} \quad P \vdash_{\leq n} \psi \quad \text{imply} \quad P \vdash_{\leq q(m+n)} \varphi \wedge \psi$$

for all formulas φ and ψ.

We can classify properties of proof systems like those above along the following lines. Some properties are *monotone* in the sense that they are preserved from weaker to stronger systems, i.e. if $P \leq Q$ and P has the property, then also Q satisfies the property. Evaluation of formulas without variables is such a monotone property. Other properties might not be monotone but still *robust* under $\leq$ in the sense that the property is preserved when we change to a $\leq$-equivalent system. Since we are interested in the degree of a proof system and not in the particular representative of that degree it would be desirable to investigate only robust or even monotone properties. But we will also see examples of properties that are *fragile* in that there exists a proof system which has the property while an equivalent system fails to satisfy this property.

The next proposition classifies the above properties according to this terminology.

Proposition 2.6.5 *1. Evaluation of formulas without variables is monotone.*

2. The following properties are $\leq$-robust: closure under modus ponens, closure under substitutions, closure under substitutions by constants, closure under disjunctions and closure under conjunctions.

3. The efficient versions of the properties from item 2 are $\leq_p$-robust.

4. Closure under multiple applications of modus ponens is fragile.

Proof. As an example for items 1 to 3 we show the robustness of modus ponens under $\leq$. Assume that P is closed under modus ponens and let p be the polynomial from the definition of closure under modus ponens. Let Q be a proof system with $P \equiv Q$ and let q_1 and q_2 be the polynomials from

$P \leq Q$ and $Q \leq P$, respectively. If $Q \vdash_{\leq m} \varphi$ and $Q \vdash_{\leq n} \varphi \to \psi$, then $P \vdash_{\leq q_2(m)} \varphi$ and $P \vdash_{\leq q_2(n)} \varphi \to \psi$. By closure of P under modus ponens we have $P \vdash_{\leq p(q_2(m)+q_2(n))} \psi$ and by $P \leq Q$ we get $Q \vdash_{\leq q_1(p(q_2(m)+q_2(n)))} \psi$.

Now we prove part 4. Let P be a proof system that is closed under multiple applications of modus ponens. For example we can choose P as a Frege system. Let φ_n and ψ_n be polynomial time constructible sequences of tautologies of strictly increasing lengths. Let p be a polynomial majorizing $|\psi_n|$ and the minimal lengths of P-proofs of φ_n and $\varphi_n \to \psi_n$. Such sequences φ_n and ψ_n are easy to find.

Now we define the system Q as

$$Q(\pi) = \begin{cases} \theta & \text{if } \pi = 0\pi',\ P(\pi') = \theta \text{ and} \\ & \theta \text{ does not appear in the sequence } \psi_n,\ n \geq 1 \\ \psi_n & \pi = 1^{4p(n)} \\ \top & \text{otherwise.} \end{cases}$$

Apparently the systems P and Q are $\leq$-equivalent. However, Q is not closed under multiple applications of modus ponens, because for each constant c we can find an n such that

$$Q \not\vdash_{\leq 2p(n)+|\psi_n|+c} \psi_n$$

because the proof length of ψ_n in Q is exactly $4p(n)$. On the other hand we have $Q \vdash_{\leq p(n)+1} \varphi_n$ and $Q \vdash_{\leq p(n)+1} \varphi_n \to \psi_n$, and hence closure under multiple applications of modus ponens fails for Q. □

We will now examine the closure properties of our standard examples of proof systems. We start with the extended Frege system which has very good closure properties.

Proposition 2.6.6 *The extended Frege system EF is efficiently closed under multiple applications of modus ponens and under substitutions. Further, it is closed under conjunctions and disjunctions.*

Proof. Modus ponens is available as a rule in EF, hence we have closure under multiple applications of modus ponens.

For closure under substitutions let $\varphi_1, \ldots, \varphi_k$ be an EF-proof of size $\leq m$. We may assume that the first l formulas in this proof are the only formulas which are derived by the extension rule, i.e. $\varphi_i = q_i \leftrightarrow \psi_i$ with extension variables q_i for $i = 1, \ldots, l$. If σ is a substitution, then

$$q_1 \leftrightarrow \sigma(\psi_1),\ \ldots\ , q_l \leftrightarrow \sigma(\psi_l),\ \sigma(\varphi_{l+1}),\ \ldots\ , \sigma(\varphi_k)$$

is an EF-proof of $\sigma(\varphi_k)$ of size $\leq m|\sigma(\varphi_k)|$.

Closure under conjunctions is achieved by applying the Frege axioms $p_1 \wedge p_2 \to p_1$, $p_1 \wedge p_2 \to p_2$ and $p_1 \to p_2 \to p_1 \wedge p_2$ together with modus ponens. Closure under disjunctions follows analogously. □

The same proposition is also valid for extensions of $EF + \Phi$ by polynomial time computable sets of axioms $\Phi \subseteq \mathrm{TAUT}$.

For resolution the situation is a bit more delicate as resolution operates only with clauses which means that it can only prove formulas in disjunctive normal form. To obtain a proof system for all tautologies we combine resolution with the truth-table system as explained in Sect. 2.2. Showing closure properties for such hybrid proof systems requires an analysis of both components. In the next proposition we do this for the truth-table method.

Proposition 2.6.7 *The truth-table system is efficiently closed under substitutions by constants and multiple applications of modus ponens. It is also closed under conjunctions, but not under disjunctions and substitutions.*

Proof. The truth-table system is closed under substitutions by constants, multiple applications of modus ponens and under conjunctions because the number of variables and hence the proof size in the truth-table system does not increase under these operations. This, however, is not the case for substitutions and disjunctions. Let φ_n be a sequence of propositional formulas such that φ_n uses n different variables. If we choose substitutions σ_n such that $\sigma_n(\varphi_n)$ has size $|\varphi_n|^{O(1)}$ and n^2 variables, then the proof size increases from 2^n for φ_n to 2^{n^2} for $\sigma_n(\varphi_n)$ which is super polynomial. Closure under disjunctions fails, for example, if we go from φ_n to $\varphi_n \vee \sigma_n(\varphi_n)$. □

For the resolution system we obtain the following closure properties:

Proposition 2.6.8 *Resolution considered as a proof system for formulas in DNF is efficiently closed under substitutions by constants, disjunctions and multiple applications of modus ponens.*

The hybrid proof system Res formed from resolution and the truth-table system is efficiently closed under substitutions by constants and multiple applications of modus ponens.

Proof. Let φ be a formula in disjunctive normal form and let σ be a substitution by constants. Hitting each clause in a resolution refutation of $\neg\varphi$ by σ we can easily transform the resulting sequence of clauses into a correct resolution refutation. Hence we obtain a refutation of the clauses corresponding to $\neg\sigma(\varphi)$.

For the case of modus ponens let Γ and Δ be sets of clauses corresponding to DNF-formulas φ and ψ, respectively. By hypothesis we have a resolution

proof of φ, i.e. Γ has a resolution refutation. Proving $\varphi \to \psi$ means that we have a resolution derivation of Γ from the clauses of Δ. Combining these two resolution proofs we refute the set Δ, i.e. ψ is proven.

For closure under disjunctions it is sufficient to observe that transforming φ into $\varphi \vee \psi$ for propositional formulas φ and ψ in DNF increases the corresponding sets of clauses, hence the formula $\varphi \vee \psi$ has the same resolution proof as φ.

As efficient closure under substitutions by constants and multiple applications of modus ponens hold for the truth-table system as well as for the resolution calculus we get them for the hybrid system defined from resolution for the set of all tautologies. □

Chapter 3

Arithmetic Theories and Propositional Proof Systems

Die Kunst beschäftigt sich mit dem Schweren und dem Guten.

Johann Wolfgang Goethe

Bounded arithmetic is closely related to propositional proof systems and disjoint NP-pairs. In this chapter we develop the general correspondence between propositional proof systems and arithmetic theories as defined by Krajíček and Pudlák (KP90).

3.1 Theories of Bounded Arithmetic

There is a number of different languages for arithmetic theories of which a detailed picture is given in (HP93). Here we will only consider the language L introduced by Buss (Bus86) which in addition to the usual ingredients $0, S, +, *, \leq$ contains a number of technical symbols in order to simplify the formalization of syntactic notions with arithmetic formulas.

The language L of arithmetic uses the symbols

$$0,\ S,\ +,\ *,\ |.|,\ \lfloor \tfrac{1}{2}.\rfloor,\ \sharp \text{ and } \leq \ .$$

$0,\ S,\ +,\ *,\ \lfloor \frac{1}{2}.\rfloor$ and $\leq$ are interpreted in the usual way. The intended interpretation of $|x|$ is $\lceil \log_2(x+1) \rceil$, i.e. the number of bits of the binary representation of x, and the smash function $x\sharp y$ is interpreted by $2^{|x|*|y|}$.

Quantifiers of the form

$$(\forall x \leq t(y)) \ldots$$

abbreviating $(\forall x)\, x \leq t(y) \rightarrow \ldots$ and

$$(\exists x \leq t(y)) \ldots$$

abbreviating $(\exists x)\, x \leq t(y) \wedge \ldots$ with some L-term t not containing the variable x are called *bounded quantifiers*. Because the function symbol $\sharp$ is included in the language and in the intended interpretation the smash function $\sharp$ has super-polynomial growth rate, that admits exactly polynomial growth in the length of the number, these bounded quantifiers can range over numbers y of length polynomial in the length of x, i.e. over exponentially large sets measured in $|x|$. If the term t is even of the form $t(y) = |s(y)|$ for some term $s(y)$, then the quantifiers are called *sharply bounded*.

Bounded L-formulas are formulas in the language of L containing only bounded quantifiers. As usual one defines a hierarchy of first-order formulas by counting their quantifier alternations. Doing this for bounded formulas we count the number of alternations of bounded quantifiers of bounded L-formulas in prenex normal form but ignoring quantifiers which are sharply bounded. The first level of this hierarchy is formed by L-formulas containing only sharply bounded quantifiers. These formulas are denoted by Σ_0^b. In the following we are particularly interested in Π_1^b- and Σ_1^b-formulas which are L-formulas in prenex normal form with only bounded universal and bounded existential quantifiers are allowed, respectively. Using a pairing function quantifiers of the same type can be combined and hence a Π_1^b-formula can be assumed to be of the form

$$(\forall y \leq t(x))\, \varphi(x, y)$$

where φ contains only sharply bounded quantifiers. Similarly, Σ_1^b-formulas look like

$$(\exists y \leq t(x))\, \varphi(x, y) \ .$$

The formula $\varphi(x, y)$ contains only sharply bounded quantifiers which range over sets of numbers of polynomial size measured in the length of x. Furthermore φ can make use of all number theoretic functions available in L. As all these functions are easy to compute $\varphi(x, y)$ can be evaluated in polynomial time for given numbers x and y. Because the existential quantifier $\exists y \leq t(x)$ can be thought of as a suitable polynomial size witness corresponding to the input x a Σ_1^b-formula describes an NP-set of natural numbers. But also all NP-sets can be defined by Σ_1^b-formulas as the next theorem which is a variant of a result of Wrathall (Wra78) (see e.g. (Kra95)) shows.

Theorem 3.1.1 *Let $\mathcal{N}$ denote the standard model of natural numbers. The subsets of $\mathcal{N}$ definable by Σ_1^b-formulas are exactly the NP-sets. Similarly,*

the subsets of $\mathcal{N}$ definable by Π_1^b-formulas equal the set of all coNP*-sets of natural numbers.*

Actually, this correspondence extends to all bounded formulas and sets from the polynomial hierarchy but we will only need it for Σ_1^b- and Π_1^b-formulas.

Given an L-theory T we say that a formula φ is a *Δ_1^b-formula with respect to T* if there exist a Σ_1^b-formula ψ and a Π_1^b-formula θ such that

$$T \vdash \varphi \leftrightarrow \psi \quad \text{and} \quad T \vdash \varphi \leftrightarrow \theta \ .$$

There is a long history of studying fragments of Peano arithmetic (see e.g. (HP93)). The fragment we need here is the theory S_2^1 introduced by Buss (Bus86). The theory is axiomatized by a finite set $BASIC$ of axioms describing the interplay of the interpretations of the function symbols S, $+$, $*$, $|.|$, $\lfloor \frac{1}{2}.\rfloor$, $\sharp$, the relation symbol $\leq$ and the constant 0. Like usual a controlled amount of induction is added to these basic axioms. In this case a version LIND of the induction scheme for the length of numbers is added:

$$\varphi(0) \wedge (\forall x)(\varphi(x) \rightarrow \varphi(x+1)) \rightarrow (\forall x)\varphi(|x|) \ .$$

Instead of this LIND-scheme it is also possible to use the polynomial induction scheme PIND which is defined as

$$\varphi(0) \wedge (\forall x)(\varphi(\lfloor \frac{x}{2} \rfloor) \rightarrow \varphi(x)) \rightarrow (\forall x)\varphi(x) \ .$$

The theory S_2^1 is then defined as the axiom set $BASIC$ augmented by the induction scheme LIND for all Σ_1^b-formulas. Equivalently, S_2^1 can be characterized as

$$S_2^1 \ = \ BASIC + \Pi_1^b\text{–LIND}$$

and

$$S_2^1 \ = \ BASIC + \Sigma_1^b\text{–PIND} \ = \ BASIC + \Pi_1^b\text{–PIND} \ .$$

The index 2 in S_2^1 refers to the presence of the function symbol $\sharp$ in the language which allows a smooth formalization of coding of sequences. This is needed for the formalization of proof systems and polynomial time computations in S_2^1. The superscript 1 in S_2^1 indicates that LIND for Σ_1^b-formulas is available in the theory. Adding Σ_i^b-LIND to $BASIC$ defines the theories S_2^i.

A central result for the theory S_2^1 is the witnessing theorem of Buss (Bus86). It describes that the proof-theoretic strength of S_2^1 corresponds to the polynomial time computable functions.

Theorem 3.1.2 (Buss (Bus86)) *Let $\varphi(x, y)$ be a Σ_1^b-formula and let*

$$S_2^1 \vdash (\forall x)(\exists y)\varphi(x, y) \ .$$

Then there exists a polynomial time computable function f which for every natural number x computes a corresponding witness y, i.e.

$$\mathcal{N} \models (\forall x)\varphi(x, f(x)) \ .$$

3.2 A Translation of Arithmetic Formulas into Propositional Formulas

To explain the connection between bounded arithmetic and propositional proof systems we have to translate first-order formulas into propositional formulas. There are essentially two translations from arithmetic formulas into propositional formulas: one was introduced by Paris and Wilkie (PW85) to transform bounded formulas in the language of $I\Delta_0$ with one extra predicate into propositional logic. The other translation dates back to Cook (Coo75) and was later adapted by Krajíček and Pudlák (KP90) to translate L-formulas into sequences of quantified propositional formulas.

We will now describe this second translation in detail. But because we do not consider quantified propositional formulas we will only explain the part of the translation which does not produce bounded quantifiers.

For L-terms t and bounded L-formulas φ we define inductively *bounding polynomials* q_t and q_φ, such that when substituting numbers of length $\leq n$ for the free variables of t or φ the evaluation of t and φ does not refer to numbers of length $> q_t(n)$ or $> q_\varphi(n)$, respectively. *Bounding polynomials for L-terms* are inductively defined as follows:

1. $q_0(n) = 1$ for all n,
2. $q_x(n) = n$ for a first-order variable x,
3. $q_{S(t)} = q_t + 1$ where t is an L-term,
4. $q_{s+t} = q_s + q_t$ for L-terms s, t,
5. $q_{s\sharp t} = q_s q_t + 1$ for L-terms s, t and
6. $q_{|t|} = q_{\lfloor \frac{t}{2} \rfloor} = q_t$ for an L-term t.

Using these bounding polynomials for terms we define inductively *bounding polynomials for bounded L-formulas*:

1. $q_{s\leq t} = q_{s=t} = q_s + q_t$ for L-terms s, t,
2. $q_{\neg\varphi} = q_\varphi$ for a bounded L-formula φ,
3. $q_{\varphi\wedge\psi} = q_{\varphi\vee\psi} = q_{\varphi\to\psi} = q_{\varphi\leftrightarrow\psi} = q_\varphi + q_\psi$ for L-formulas φ, ψ and
4. $q_{(\forall x\leq t)\varphi}(n) = q_{(\exists x\leq t)\varphi}(n) = q_t(n)+q_\varphi(n+q_t(n))$ for a bounded L-formula φ and an L-term t.

Let $\|+\|_m$, $\|*\|_m$, $\|\lfloor\frac{1}{2}.\rfloor\|_m$, $\||.|\|_m$ and $\|\sharp\|_m$ be m-tupels of polynomial size boolean formulas computing the first m bits of the corresponding functions on inputs of length m.

For each L-term t we now define for $m > q_t(n)$ an m-tupel $\|t\|^n_m$ of propositional formulas. For every free variable x in t we introduce a sequence $p^x_{n-1}, \ldots, p^x_0$ of propositional variables which represent the values of the bits of x where p^x_0 takes the value of the least significant bit. By induction on the logical complexity of terms t we define m-tupels of propositional formulas $\|t\|^n_m$ which compute the first m bits of the value of t for inputs of length $\leq n$:

1. $\|0\|^n_m$ is the m-tupel $(\bot, \ldots, \bot)$.
2. For a variable x we set $\|x\|^n_m = (\bot, \ldots, \bot, p^x_{n-1}, \ldots, p^x_0)$ with $m - n$ leading $\bot$.
3. $\|s + t\|^n_m = \| + \|_m (\|s\|^n_m, \|t\|^n_m)$ for L-terms s and t and
4. analogously for the other L-functions.

An L-formula φ is in *negation implication normal form* (NINF) if φ is in prenex normal form and does not contain the connectives $\to$ or $\leftrightarrow$, and negations occur only directly before atomic formulas. To a formula φ in NINF we assign special propositional variables $\nu^\varphi_0, \nu^\varphi_1, \ldots$ called the universal variables of φ and propositional variables $\varepsilon^\varphi_0, \varepsilon^\varphi_1, \ldots$ called the existential variables of φ.

For Σ^b_1- and Π^b_1-formulas φ in NINF we define by induction on the logical complexity of φ propositional translations $\|\varphi\|^n_m$ for $m \geq q_\varphi(n)$. The translation can be extended to Σ^b_1- and Π^b_1-formulas which are not in NINF by transforming these formulas into NINF. The translation is defined as follows.

1. $\|s = t\|^n_m = EQ_m (\|s\|^n_m, \|t\|^n_m)$
   with $EQ_m (\bar{p}, \bar{q}) = \bigwedge_{i=0}^{m-1} p_i \leftrightarrow q_i$
2. $\|s \leq t\|^n_m = LE_m (\|s\|^n_m, \|t\|^n_m)$
   with $LE_m (\bar{p}, \bar{q}) = \bigvee_{i=0}^{m-1} \left(\left(\bigwedge_{j=i+1}^{m-1} p_j \leftrightarrow q_j\right) \wedge \neg p_i \wedge q_i\right) \vee EQ_m (\bar{p}, \bar{q})$

3. $\|\neg\varphi\|^n_m = \neg\|\varphi\|^n_m$ for atomic formulas φ.

4. $\|\varphi \wedge \psi\|^n_m = \|\varphi\|^n_m \wedge \|\psi\|^n_m$

5. $\|\varphi \vee \psi\|^n_m = \|\varphi\|^n_m \vee \|\psi\|^n_m$

6. $\|\,(\forall x \leq t)\,\varphi(x)\|^n_m = \|\neg(x \leq t) \vee \varphi(x)\|^n_m\,(p^x_i/\nu^\varphi_i)^{m-1}_{i=0}$,
where the term t is not of the form $|s|$. The suffix $(p^x_i/\nu^\varphi_i)^{m-1}_{i=0}$ indicates that the variables $p^x_{m-1},\ldots,p^x_0$ are replaced by the universal variables $\nu^\varphi_{m-1},\ldots,\nu^\varphi_0$. This is necessary for the case that φ contains several universal quantifications over x.

7. $\|\,(\exists x \leq t)\,\varphi(x)\|^n_m = \|(x \leq t) \wedge \varphi(x)\|^n_m\,(p^x_i/\varepsilon^\varphi_i)^{m-1}_{i=0}$,
where the term t is not of the form $|s|$.
Again, the substitution $(p^x_i/\varepsilon^\varphi_i)^{m-1}_{i=0}$ is necessary because the formula that we want to translate might contain more than one existential quantification over x. But as these different existential quantifiers are usually not witnessed by the same element we need different propositional variables for each quantifier.

8. $\|\,(\forall x \leq |t|)\,\varphi(x)\|^n_m = \bigwedge_{k=0}^{m-1} \|\neg\,(\underline{k} \leq |t|) \vee \varphi\,(\underline{k})\,\|^n_m$, where $\underline{k}$ is some dyadic representation of the natural number k.

9. $\|\,(\exists x \leq |t|)\,\varphi(x)\|^n_m = \bigvee_{k=0}^{m-1} \|\underline{k} \leq |t| \wedge \varphi\,(\underline{k})\,\|^n_m$

In the following we will omit the explicit reference to the bounding polynomial and write simply $\|\varphi\|^n$ in place of $\|\varphi\|^n_{q(n)}$. Abbreviating further we will also use $\|\varphi(x)\|$ to denote the set $\{\|\varphi(x)\|^n \,|\, n \geq 0\}$. We will also usually associate first-order formulas $\varphi(\bar{x})$ with free variables with their universally closed counterparts $(\forall\bar{x})\varphi(\bar{x})$. Therefore the above translation is not only suitable for Π^b_1- but in fact for $\forall\Pi^b_1$-formulas.

The formula $\|\varphi(x)\|^n$ has n propositional variables $p^x_{n-1},\ldots,p^x_0$ corresponding to the bits of x. If $\varphi(x) = (\forall y \leq t)\psi(x,y)$ is a Π^b_1-formula, then additionally the universal variables $\nu^\psi_0, \nu^\psi_1, \ldots$ occur in $\|\varphi(x)\|^n$. If $a \in \mathcal{N}$ is a number of length $\leq n$ we denote the bits of a by $\bar{a}$. Substituting $p^x_{n-1},\ldots,p^x_0$ by the constants $\bar{a}$ we arrive at formulas $\|\varphi(x)\|^n(\bar{p}^x/\bar{a})$ with only the universal variables $\nu^\varphi_0, \nu^\varphi_1, \ldots$ remaining free. These formulas provide a precise description of the truth value of $\varphi(a)$. We state this in the next theorem which is essentially due to Cook (Coo75). Its proof is immediate from the construction of the translations $\|.\|$.

Theorem 3.2.1 (Cook (Coo75)) *1. For $\varphi \in \Pi^b_1$ or $\varphi \in \Sigma^b_1$ the sequence $\|\varphi\|^n = \|\varphi\|^n_{q(n)}$ consists of propositional formulas which have*

polynomial size in n. Moreover, the sequence $\|\varphi\|^n$ is polynomial time constructible, i.e. there exists a polynomial time computable algorithm that on input 1^n outputs the formula $\|\varphi\|^n$.

2. *The sequence $\|\varphi\|^n$ is a propositional description of the first-order formula φ, more precisely:*

   (a) *If $\varphi(x) \in \Pi_1^b$, then for all $a \in \mathcal{N}$ with $|a| \leq n$ the formula $\|\varphi(x)\|^n(\bar{p}^x/\bar{a})$ is a tautology if and only if $\mathcal{N} \models \varphi(a)$. In particular, the formula $\|\varphi(x)\|^n$ is a tautology if and only if $\varphi(a)$ holds for all natural numbers a of length $\leq n$.*

   (b) *If $\varphi(x) \in \Sigma_1^b$, then for all $a \in \mathcal{N}$ with $|a| \leq n$ the formula $\|\varphi(x)\|^n(\bar{p}^x/\bar{a})$ is satisfiable if and only if $\mathcal{N} \models \varphi(a)$.*

3.3 Coding Propositional Proofs in Bounded Arithmetic

In order to formalize concepts such as propositional proof systems in L-theories it is necessary to define polynomial time computations with L-formulas. As the language L was suitably chosen to include the technical symbols $|.|$, $\lfloor \frac{1}{2}. \rfloor$ and $\sharp$ it is relatively easy to define a pairing function and a coding of finite sets and sequences. Using this it is possible to code descriptions of Turing machine computations. In particular using the length induction scheme LIND the theory S_2^1 can prove the uniqueness of suitably encoded polynomial time computations, i.e. S_2^1 proves that for all polynomial time deterministic Turing machines M and all inputs x there exists exactly one computation of $M(x)$. Expressed differently, polynomial time computations are Δ_1^b-definable in S_2^1. This is described in detail in Chap. V of (HP93) and Chap. 6 of (Kra95).

Encoding propositional formulas as numbers in some straightforward way we can in a theory T speak of propositional formulas, assignments and proofs. Instead of giving the details of the encoding we will just introduce some notation (similar as in (Kra95)). A more detailed description of these concepts can be found in (Bus98a).

First we need to encode propositional formulas as numbers. Let

$$Form$$

be a Σ_0^b-formula such that $\mathcal{N} \models Form(\varphi)$ if and only if φ is the encoding of a propositional formula. Let

$$Assign(\alpha, \varphi)$$

be a Σ_0^b-formula describing that α is the encoding of an assignment of the variables of the propositional formula encoded by φ. Similarly, let the Σ_0^b-formula

$$Eval(\alpha, \varphi, \gamma)$$

describe that γ is an evaluation of the propositional formula φ under the assignment α. By

$$\alpha \models \varphi$$

we denote a first-order description for the fact that α is a satisfying assignment for the formula φ. Using the earlier definitions $\alpha \models \varphi$ can be expressed as

$$(\exists\gamma)Eval(\alpha, \varphi, \gamma) \wedge \varphi(\gamma) = 1 \ .$$

Since the length of γ can be bounded by a polynomial in the length of φ, this is a Σ_1^b-formula. In the following we will always assume that quantifiers such as $\exists\gamma$ above are implicitly bounded by the quantified formulas. Because the evaluation γ of the formula φ is unique and this uniqueness is provable in S_2^1, i.e.

$$S_2^1 \vdash Eval(\alpha, \varphi, \gamma_1) \wedge Eval(\alpha, \varphi, \gamma_2) \rightarrow \gamma_1 = \gamma_2$$

it follows that

$$(\forall\gamma)Eval(\alpha, \varphi, \gamma) \rightarrow \varphi(\gamma) = 1$$

is a Π_1^b-definition of $\alpha \models \varphi$ which is in S_2^1 provably equivalent to the above Σ_1^b-definition, hence $\alpha \models \varphi$ is Δ_1^b with respect to S_2^1 (see (Kra95) Sect. 9.3 for the details).

Now we are ready to formalize tautologies. For this let $\mathrm{Taut}(\varphi)$ be an L-formula asserting that all assignments satisfy the formula φ, i.e.

$$(\forall\alpha)Assign(\alpha, \varphi) \rightarrow \alpha \models \varphi \ .$$

Because $\alpha \models \varphi$ has a Π_1^b-definition and $Assign$ is a Σ_0^b-formula this definition of Taut is a Π_1^b-formula.

Finally we need to code propositional proofs. For a propositional proof system P let

$$\mathrm{Prf}_P(\pi, \varphi)$$

be an L-formula describing that π is the encoding of a correct P-proof of the propositional formula encoded by φ. Because P is a polynomial time computable function Prf_P is definable by a Σ_1^b-formula. But like all polynomial time computable functions the predicate Prf_P also has a Π_1^b-definition. Moreover, these definitions can be chosen in such a way that the theory S_2^1 proves their equivalence, hence Prf_P is Δ_1^b-definable with respect to S_2^1.

3.4 Consistency Statements

The consistency of a proof system is described by the *consistency statement* of a proof system

$$\mathrm{Con}(P) = (\forall \pi) \neg \mathrm{Prf}_P(\pi, \bot) \ .$$

A somewhat stronger formulation of consistency is given by the *reflection principle* of a propositional proof system P which is defined by the L-formula

$$\mathrm{RFN}(P) = (\forall \pi)(\forall \varphi)\mathrm{Prf}_P(\pi, \varphi) \rightarrow \mathrm{Taut}(\varphi) \ .$$

From the remarks in the previous section it follows that $\mathrm{Con}(P)$ and $\mathrm{RFN}(P)$ are $\forall \Pi^b_1$-formulas.

These two consistency notions are compared by the following well known observation, contained e.g. in (Kra95):

Proposition 3.4.1 *Let P be a proof system that is closed under substitutions by constants and modus ponens and evaluates formulas without variables. Assume further that these properties are provable in S^1_2. Then*

$$S^1_2 \vdash \mathrm{RFN}(P) \leftrightarrow \mathrm{Con}(P) \ .$$

Proof. Suppose $S^1_2 \vdash \mathrm{RFN}(P)$. This means in particular that

$$S^1_2 \vdash (\forall \pi)\mathrm{Prf}_P(\pi, \bot) \rightarrow \mathrm{Taut}(\bot) \ .$$

Because $\mathrm{Taut}(\bot)$ is false in S^1_2 this implies

$$S^1_2 \vdash (\forall \pi) \neg \mathrm{Prf}_P(\pi, \bot)$$

which means $S^1_2 \vdash \mathrm{Con}(P)$.

For the opposite implication assume that $S^1_2 \nvdash \mathrm{RFN}(P)$. Hence there exists a model M of S^1_2 and a propositional formula $\varphi(\bar{p})$ such that

$$M \models (\exists \pi)\mathrm{Prf}_P(\pi, \varphi(\bar{p})) \wedge \neg \mathrm{Taut}(\varphi(\bar{p})) \ .$$

This means that there exists an assignment α such that

$$M \models (\exists \pi)\mathrm{Prf}_P(\pi, \varphi(\bar{p})) \wedge \alpha \not\models \varphi(\bar{p}) \ .$$

Let α map the variables $\bar{p}$ of $\varphi(\bar{p})$ to the tuple $\bar{a}$. Hence $\varphi(\bar{a})$ is a false formula without variables. By assumption S^1_2 proves that $\neg\varphi(\bar{a})$ is provable in P. Because P is provably closed under substitutions by constants we get

$$M \models (\exists \pi)\mathrm{Prf}_P(\pi, \varphi(\bar{a})) \wedge (\exists \pi')\mathrm{Prf}_P(\pi', \neg\varphi(\bar{a})) \ .$$

By closure of P under modus ponens in S^1_2 we obtain

$$M \models (\exists \pi)\mathrm{Prf}_P(\pi, \bot) \ .$$

Hence $\mathrm{Con}(P)$ fails in M and as $M \models S^1_2$ the theory S^1_2 does not prove the consistency principle of P. □

Very often we will consider propositional descriptions of the reflection principle. These can be simply obtained by translating $\mathrm{RFN}(P)$ to a sequence of propositional formulas using the translation $\|.\|$:

Definition 3.4.2 *A propositional proof system P has the* reflection property *if*

$$P \vdash_* \|\mathrm{RFN}(P)\|^n \ .$$

At some places we need the more efficient version of this definition that short P-proofs of $\|\mathrm{RFN}(P)\|^n$ are constructible.

Definition 3.4.3 *We say that a propositional proof system P has the* strong reflection property *if there exists a polynomial time algorithm that on input 1^n outputs a P-proof of $\|\mathrm{RFN}(P)\|^n$.*

There is a subtle problem with Definitions 3.4.2 and 3.4.3 that is somewhat hidden in the definitions. Namely, the formula $\mathrm{Prf_P}$ describes the computation of some Turing machine computing the function P. However, the provability of the formulas $\|\mathrm{RFN}(P)\|^n$ with polynomial size P-proofs might depend on the actual choice of the Turing machine computing P. We will illustrate this by an example which unfortunately has to be postponed until Sect. 3.8 (Proposition 3.8.3). Nevertheless, this observation tells us that we should understand the meaning of Definition 3.4.2 in the following, more precise way: a propositional proof system P has the reflection property if there exists a deterministic polynomial time Turing machine M computing the function P such that for a suitable Δ^b_1-formalization Prf_P of the computation of M with respect to S^1_2 we have

$$P \ \vdash_* \ \|(\forall \pi)(\forall \varphi)\mathrm{Prf}_P(\pi, \varphi) \to \mathrm{Taut}(\varphi)\|^n \ .$$

The same applies to Definition 3.4.3.

3.5 The Correspondence Between Arithmetic Theories and Propositional Proof Systems

Krajíček and Pudlák introduced in (KP90) a general correspondence between L-theories T and propositional proof systems P. Pairs (T, P) from this correspondence possess in particular the following two properties:

1. For all $\varphi(x) \in \Pi_1^b$ with $T \vdash (\forall x)\varphi(x)$ we have $P \vdash_* \|\varphi(x)\|^n$.

2. T proves the correctness of P, i.e. $T \vdash \mathrm{RFN}(P)$. Furthermore P is the strongest proof system for which T proves the correctness, i.e. $T \vdash \mathrm{RFN}(Q)$ for a proof system Q implies $Q \leq P$.

Actually, (KP90) contains a stronger formulation, namely properties 1 and 2 are required to be provable in S_2^1. The properties 1 and 2 then take the following form:

3. For all $\varphi(x) \in \Pi_1^b$ with $T \vdash (\forall x)\varphi(x)$ we have

$$S_2^1 \vdash (\forall n)(\exists \pi_n)\mathrm{Prf}_P(\pi_n, \|\varphi(x)\|^{|n|}) \ .$$

4. T proves the correctness of P, i.e. $T \vdash \mathrm{RFN}(P)$.

From Buss' witnessing theorem for S_2^1 (Theorem 3.1.2) it follows that a proof π_n of $\|\varphi(x)\|^{|n|}$ can be computed in polynomial time from the number n. Therefore condition 3 implies condition 1.

It is then even possible to derive the second part of property 2 as a consequence of 3 and 4 (cf. (Pud98)), i.e. if T and P fulfill the conditions 3 and 4, then every proof system Q with $T \vdash \mathrm{RFN}(Q)$ is p-simulated by P, and this p-simulation is provable in S_2^1. In contrast we only stated the weak simulation $Q \leq P$ in condition 2.

For our purpose conditions 1 and 2 are mostly sufficient. Therefore we make the following definition:

Definition 3.5.1 *A propositional proof system P is called* regular *if there exists an L-theory T such that properties 1 and 2 are fulfilled for (T, P).*

Occasionally, we will also need a strengthened version of regularity, but still weaker than properties 3 and 4.

Definition 3.5.2 *We call a propositional proof system P* strongly regular *if there exists an L-theory T such that the following two properties are fulfilled for (T, P).*

5. *Let $\varphi(x)$ be a Π_1^b-formula such that $T \vdash (\forall x)\varphi(x)$. Then there exists a polynomial time computable function f that on input 1^n outputs a P-proof of $\|\varphi(x)\|^n$.*

6. *$T \vdash \mathrm{RFN}(P)$ and if $T \vdash \mathrm{RFN}(Q)$ for some proof system Q, then $Q \leq_p P$.*

In comparison to regularity conditions 1 and 2 we gave these axioms a constructive formulation: in 5 P-proofs are polynomial time constructible and in 6 we have p-simulations instead of $\leq$. Clearly, conditions 3 and 4 imply the strong regularity conditions 5 and 6 which in turn imply the regularity conditions 1 and 2.

In Sect. 3.7 we will discuss sufficient conditions for the regularity and strong regularity of propositional proof systems.

If T is an L-theory such that there exists a regular proof system P satisfying conditions 1 and 2, then P is unique up to $\leq$-equivalence by property 2. Conversely, if P is a proof system for which there exists an L-theory T satisfying conditions 3 and 4, then the $\forall\Pi^b_1$-consequences of T are determined by P. This is the contents of the next theorem which is essentially contained in (KP90).

Theorem 3.5.3 *1. Let T be an L-theory and P_1, P_2 be proof systems such that both (T, P_1) and (T, P_2) satisfy conditions 1 and 2. Then $P_1 \equiv P_2$.*

2. Let $T \supseteq S^1_2$ be an L-theory and P a proof system such that conditions 3 and 4 are satisfied for (T, P). Then the theories T and $S^1_2 + \mathrm{RFN}(P)$ have the same set of $\forall\Pi^b_1$-consequences.

Proof. Part 1 follows immediately from condition 2 for (T, P_1) and (T, P_2).

For part 2 let T be an extension of S^1_2 and P a proof system such that conditions 3 and 4 hold. As $S^1_2 \subseteq T$ and $T \vdash \mathrm{RFN}(P)$ all $\forall\Pi^b_1$-consequences of $S^1_2 + \mathrm{RFN}(P)$ are also provable in T.

For the other inclusion let $\varphi(x)$ be a Π^b_1-formula such that

$$T \vdash (\forall x)\varphi(x) \ .$$

By condition 3 this implies

$$S^1_2 \vdash (\forall n)(\exists \pi_n)\mathrm{Prf}_P(\pi_n, \|\varphi(x)\|^{|n|}) \ .$$

Using the reflection principle of P we infer

$$S^1_2 + \mathrm{RFN}(P) \vdash (\forall n)\mathrm{Taut}(\|\varphi(x)\|^{|n|}) \ .$$

By induction on the logical complexity of φ we can show

$$S^1_2 \vdash (\forall n)\mathrm{Taut}(\|\varphi(x)\|^{|n|}) \to (\forall x)(|x| \leq |n| \to \varphi(x))$$

and hence we obtain $S^1_2 + \mathrm{RFN}(P) \vdash (\forall x)\varphi(x)$. □

Before we continue the investigation of regular systems we will give an informal discussion on the properties of the correspondence between arithmetic theories and propositional proof systems.

Part 1 of the correspondence is called the *simulation of T by P*. Its main application is the uniform construction of P-proofs. We will explain this in some more detail. If some Π^b_1-formula φ is T-provable, then as $\mathcal{N}$ is a model of T we have in particular $\mathcal{N} \models \varphi$. Hence by Theorem 3.2.1 the sequence $\|\varphi\|^n$ contains only tautologies. But moreover by part 1 of the correspondence the tautologies of this sequence have polynomial size P-proofs. Usually these P-proofs are also constructible in polynomial time as follows. The T-proof of φ is given in some first-order sequent calculus suitable for the language L. The first-order sequent calculus proof of φ is then translated to a sequence of propositional proofs in some propositional sequent calculus which is a propositional counterpart of the first-order calculus. The translation proceeds by replacing each application of a first-order rule by an application of the corresponding propositional rule. As the first-order rules are often more flexible than their propositional versions it is necessary to fill in the gaps between the steps. If carefully done this results in a sequence of propositional proofs of polynomial size in the respective propositional calculus which then has to be transformed into a sequence of P-proofs. We will sketch this procedure for the correspondence of S^1_2 and EF in Sect. 3.6.

If one replaces condition 1 by the stronger condition 3 then P-proofs for the sequence $\|\varphi\|^n$ are always constructible in polynomial time. This follows from condition 3 because Buss' witnessing theorem applied to

$$S^1_2 \vdash (\forall n)(\exists \pi_n)\mathrm{Prf}_P(\pi_n, \|\varphi(x)\|^{|n|})$$

yields a polynomial time computable function f that on input n produces the P-proof π_n.

As it is mostly easier to show the validity of a first-order principle in some theory than to explicitely construct sequences of propositional proofs the correspondence provides an elegant method to construct short propositional proofs. Therefore theories of bounded arithmetic and propositional proof systems are often seen in analogy to the correspondence of Turing machines to Boolean circuits as the uniform and respective non-uniform realization of the same concept.

Additionally, the correspondence also allows to show lower bounds to the length of propositional proofs. This requires some model-theoretic machinery which we will describe next.

Let M be a model of $Th(\mathcal{N})$ and let $n \in M$ be a non-standard element. Then we define the cut M_n in the model M as

$$M_n = \{b \in M \mid |b| \leq n^k \text{ for some } k \in \mathcal{N}\} \ .$$

The next theorem explained in (Kra01b) offers a model-theoretic way to show lower bounds to the length of propositional proofs.

Theorem 3.5.4 *Let P be a regular proof system and let T be the theory corresponding to P. Assume further that P is closed under modus ponens and substitutions by constants, and let $\varphi(x)$ be a Π^b_1-formula. Then the following two conditions are equivalent:*

1. *For every model $M \models Th(\mathcal{N})$ and every non-standard element $a \in M \setminus \mathcal{N}$ there exists a model N such that*
   (a) *$N \supseteq M_n$ where $n = |a|$,*
   (b) *$N \models T$,*
   (c) *$N \models \neg\varphi(a)$ and*
   (d) *If $M_n \models \mathrm{Prf_P}(\pi, \psi)$ for some π, ψ, then also $N \models \mathrm{Prf_P}(\pi, \psi)$.*
2. *There does not exist a sequence of pairwise distinct natural numbers a_i, $i \in \mathcal{N}$, of length $n_i = |a_i|$ such that*

$$P \vdash_* \|\varphi(x)\|^{n_i}(\bar{p}^x/\bar{a}_i) \ .$$

Proof. For the forward implication let a_i, $i \in \mathcal{N}$ be pairwise distinct natural numbers and let $n_i = |a_i|$. Assume that $\|\varphi(x)\|^{n_i}(\bar{p}^x/\bar{a}_i)$ have P-proofs of length $\leq n_i^k$ for some $k \in \mathcal{N}$, i.e.

$$\mathcal{N} \models (\exists\pi)|\pi| \leq n_i^k \wedge \mathrm{Prf}_P(\pi, \|\varphi(x)\|^{n_i}(\bar{p}^x/\bar{a}_i)) \ .$$

By compactness there exist a model $M \models Th(\mathcal{N})$ and non-standard element $a \in M \setminus \mathcal{N}$, $|a| = n$ such that

$$M \models (\exists\pi)|\pi| \leq n^k \wedge \mathrm{Prf}_P(\pi, \|\varphi(x)\|^{n}(\bar{p}^x/\bar{a})) \ .$$

Let now N be a model satisfying the conditions 1a to 1d. Because $a, \pi \in M_n$ and

$$M_n \models \mathrm{Prf}_P(\pi, \|\varphi(x)\|^{n}(\bar{p}^x/\bar{a}))$$

we obtain with condition 1d also

$$N \models \mathrm{Prf}_P(\pi, \|\varphi(x)\|^{n}(\bar{p}^x/\bar{a})) \ .$$

$N \models T$ and $T \vdash \mathrm{RFN}(P)$ imply

$$N \models \mathrm{Taut}(\|\varphi(x)\|^n(\bar{p}^x/\bar{a})) \ .$$

On the other hand $N \models \neg\varphi(a)$ yields an assignment α such that

$$N \models (\alpha \models \neg\|\varphi(x)\|^n(\bar{p}^x/\bar{a}))$$

which gives a contradiction.

For the reverse implication let $M \models Th(\mathcal{N})$ and $a \in M \setminus \mathcal{N}$ with $|a| = n$. Assume that for all $N \supseteq M_n$, $N \models T$ we have $N \models \varphi(a)$. Then we infer

$$Diag(M_n) \cup T \vdash \varphi(a) \ ,$$

and with compactness there exists a tupel $\bar{b} \in M_n$ and formula $\psi(a, \bar{b}) \in Diag(M_n)$ such that

$$T \vdash \psi(a, \bar{b}) \to \varphi(a) \ .$$

Hence

$$T \vdash (\forall x, \bar{y})\psi(a, \bar{y}) \to \varphi(a) \ .$$

As this is a $\forall\Pi^b_1$-formula there exist polynomial size P-proofs of the formulas

$$\|\psi(x, \bar{y}) \to \varphi(x)\|^{n,\bar{m}} = \|\psi(x, \bar{y})\|^{n,\bar{m}} \to \|\varphi(\bar{p}^x)\|^n \ . \tag{3.1}$$

Because $\bar{b} \in M_n$ we have in particular $|\bar{b}| \leq |a|^k$ for some $k \in \mathcal{N}$. Therefore the P-proofs of the formulas (3.1) have proofs of size polynomial in n.

Because $M \models Th(\mathcal{N})$ and for non-standard elements $a, \bar{b}$ we have $M \models \psi(a, \bar{b})$ there exists by underspill an infinite sequence of standard elements $\mathcal{N} \models \psi(a_i, \bar{b}_i)$. As the formulas $\psi(a_i, \bar{b}_i)$ are contained in $Diag(M_a)$ their $\|.\|$-translations have polynomial size P-proofs. Because P is closed under modus ponens and substitutions by constants we get by substituting $a_i, \bar{b}_i$ into the P-proofs of the formulas (3.1) polynomial size P-proofs of the formulas $\|\varphi(x)\|^{|a_i|}(\bar{p}^x/\bar{a}_i)$. □

Part 2 of the correspondence expresses that from the knowledge of the theory T the proof system P is an optimal proof system. This can be used to show simulations between proof systems. Namely, to show $Q \leq P$ for a regular proof system P it suffices to prove $\mathrm{RFN}(Q)$ in the theory T associated with P. In this way it was shown for example that the substitution Frege system SF is simulated by the extended Frege system EF (Dow85; KP89). For this it is enough to verify that $S^1_2 \vdash \mathrm{RFN}(SF)$ which is considerably simpler than to give a direct propositional simulation (KP89).

3.6 The Correspondence Between S_2^1 and EF

In this section we will describe the correspondence between S_2^1 and EF. We start with property 1 of the correspondence which states the simulation of S_2^1 by EF. We will only sketch the proof as a complete presentation is very tedious. The theorem is essentially contained in (Coo75) but for the theory PV instead of S_2^1. A complete proof is contained in (Kra95).

Theorem 3.6.1 (Cook (Coo75), Buss (Bus86)) *Let φ be a Π_1^b-formula. Then*

$$S_2^1 \vdash (\forall \bar{x})\varphi(\bar{x}) \quad \textit{implies} \quad EF \vdash_* \|\varphi(\bar{x})\|^n \ .$$

In fact, the EF-proofs of $\|\varphi(\bar{x})\|^n$ can be constructed in polynomial time.

Proof. The proof proceeds along the following lines.

First step. We fix a first-order sequent calculus LKB which extends the propositional sequent calculus LK by rules for the introduction of quantifiers, both bounded and unbounded. An example for such a rule is

$$\frac{A(t), \Gamma \longrightarrow \Delta}{t \leq s, (\forall x \leq s)A(x), \Gamma \longrightarrow \Delta}$$

for the introduction of a bounded universal quantifier on the left side of a sequent. Additionally, for all axioms A from $BASIC$ sequents

$$\longrightarrow A$$

are introduced, and the polynomial induction scheme PIND is formalized by the inference rule

$$\frac{\Gamma, A(\lfloor \frac{a}{2} \rfloor) \longrightarrow A(a), \Delta}{\Gamma, A(0) \longrightarrow A(t), \Delta} \ .$$

where t is an arbitrary term and the variable a does not occur in the lower sequent.

The above sequent calculus is defined in such a way that for any formula B

$$S_2^1 \vdash B$$

if and only if the sequent

$$\longrightarrow B$$

has an $LKB + \Sigma_1^b$-PIND-proof from the initial sequents corresponding to $BASIC$.

Second step. Assume now that as in the hypothesis of this theorem $\varphi(\bar{x})$ is a Π_1^b-formula such that

$$S_2^1 \vdash (\forall \bar{x})\varphi(\bar{x}) \ .$$

By the first step above this means that there exists an $LKB+\Sigma_1^b$-PIND-proof π of

$$\longrightarrow (\forall x)\varphi(x)$$

from the sequents for $BASIC$.

By Gentzen's cut-elimination theorem (Gen35) adapted to the LKB-calculus (Bus86) it follows that the proof π can be chosen in such a way that all formulas occurring in π are Σ_1^b or Π_1^b.

Third step. Now we want to transform the LKB-proof π from the second step to a sequence of propositional EF-proofs. The idea of this simulation of S_2^1 by EF is to choose a bounding polynomial q that bounds all formulas in π and then translate every formula B occurring in π to $\|B\|_{q(m)}^m$. This is possible as all formulas B in π are Σ_1^b- or Π_1^b-formulas. This itself might not produce valid EF-proofs but filling the gaps by polynomial size EF-derivations results in the desired EF-proofs of $\|\varphi\|_{q(m)}^m$. We will illustrate this process by some examples. A complete presentation of this step is contained in Chapter 9 of (Kra95).

For the construction of the EF-proofs we show by induction on the number of inferences before a sequent

$$\Gamma \longrightarrow \Delta$$

from π that the propositional formulas

$$\|\neg\Gamma \vee \Delta\|$$

which is an abbreviation for

$$\bigvee_{A\in\Gamma} \|\neg A\|_{q(m)}^m \vee \bigvee_{B\in\Delta} \|B\|_{q(m)}^m$$

have EF-proofs of size polynomial in m.

The first thing to verify is that translations of initial sequents have polynomial size EF-proofs. This involves proving translations of logical axioms like

$$B \longrightarrow B$$

and translations of the axioms of $BASIC$.

Most of the structural rules like

$$\frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, B}$$

are easy to prove even in the Frege system and therefore do not present any difficulty.

For the contraction rule

$$\frac{\Gamma \longrightarrow B, B, \Delta}{\Gamma \longrightarrow B, \Delta}$$

there is the problem that different occurrences of B use different existential variables. Let the existential variables of the three occurrences of B in the above rule be ε_i, ε_i' and ε_i'', respectively. By induction hypothesis there exist polynomial size EF-proofs of

$$\|\neg\Gamma \vee B \vee B \vee \Delta\| \ .$$

We extend these EF-proofs by using the extension rule in EF

$$\varepsilon_j'' \leftrightarrow (\|B\|(\bar{\varepsilon}) \wedge \varepsilon_j) \vee (\neg\|B\|(\bar{\varepsilon}) \wedge \varepsilon_j')$$

and then derive from the hypothesis

$$\|\neg\Gamma\| \vee \|B\|(\bar{\varepsilon}) \vee \|B\|(\bar{\varepsilon}') \vee \|\Delta\|$$

the conclusion

$$\|\neg\Gamma\| \vee \|B\|(\bar{\varepsilon}'') \vee \|\Delta\| \ .$$

Nontrivial technical difficulties arise by the rules for the introduction of the quantifiers. We will not discuss this here but instead finish the induction proof by explaining how to handle the Σ_1^b-PIND-rule

$$\frac{\Gamma, A(\lfloor \frac{a}{2} \rfloor) \longrightarrow A(a), \Delta}{\Gamma, A(0) \longrightarrow A(t), \Delta} \ .$$

By induction hypothesis we have polynomial size EF-proofs of the formulas

$$\|\neg\Gamma\| \ \vee \ \|\neg A(\lfloor \frac{a}{2} \rfloor)\| \ \vee \ \|A(a)\| \ \vee \ \|\Delta\| \ .$$

As EF is efficiently closed under substitutions by constants we can construct polynomial size EF-proofs of all formulas

$$\|\neg\Gamma\| \ \vee \ \|\neg A(\lfloor \frac{a}{2} \rfloor)\|(a/2^i) \ \vee \ \|A(a)\|(a/2^i) \ \vee \ \|\Delta\| \ .$$

for all numbers $i \leq q(m)$. Then we use a series of propositional cuts to obtain EF-proofs of the $\|.\|$-translations of the following formulas

$$\frac{\Gamma, A(0) \longrightarrow A(1), \Delta \qquad \Gamma, A(1) \longrightarrow A(2), \Delta}{\Gamma, A(0) \longrightarrow A(2), \Delta} \ ,$$

from this we derive

$$\frac{\Gamma, A(0) \longrightarrow A(2), \Delta \qquad \Gamma, A(2) \longrightarrow A(4), \Delta}{\Gamma, A(0) \longrightarrow A(4), \Delta}$$

and so forth. Simulating this construction in EF results in polynomial size EF-proofs of the $\|.\|$-translations of

$$\Gamma, A(0) \longrightarrow A(t), \Delta \ .$$

□

Examining the proof of this theorem it is apparent that the theorem is still valid if both the theory S^1_2 and the proof system EF are enhanced by further axioms. In particular, to add the reflection principle of a propositional proof system will be of central interest for the following section. We formulate this version of Theorem 3.6.1 in the following corollary.

Corollary 3.6.2 *Let Φ be a polynomial time decidable set of true Π^b_1-formulas, i.e. $\mathcal{N} \models \varphi$ for all $\varphi \in \Phi$. Then the proof system $EF + \|\Phi\|$ simulates the theory $S^1_2 + \Phi$, i.e. for all Π^b_1-formulas ψ*

$$S^1_2 + \Phi \vdash (\forall \bar{x})\psi(\bar{x}) \quad \textit{implies} \quad EF + \|\Phi\| \vdash_* \|\psi(\bar{x})\|^n \ .$$

Additionally, the $EF + \|\Phi\|$-proofs of $\|\psi(\bar{x})\|^n$ can be constructed in polynomial time.

Proof. Adding the formulas Φ as axioms to the theory S^1_2 corresponds to enhancing the first-order sequent calculus LKB from the first step of the previous proof by the initial sequents

$$\longrightarrow \varphi$$

for all formulas $\varphi \in \Phi$. The transformation of these sequents into $EF + \|\Phi\|$-proofs in the third step of the last proof does not present any problem as the $\|.\|$-translations of all formulas from Φ are available in the proof system. □

Before we come to part 6 of the correspondence between S^1_2 and EF we need a technical lemma which describes that EF can evaluate the $\|.\|$-translations of the first-order formula Taut. The proof proceeds by induction on the logical complexity of formulas.

Lemma 3.6.3 (Krajíček, Pudlák (KP90)) *For all propositional formulas φ we have*

$$EF \vdash_* \|\mathrm{Taut}(\varphi)\|^{|\varphi|} \to \varphi \ .$$

Moreover, the EF-proofs of these formulas are constructible in polynomial time.

We continue with property 6 of the correspondence.

Theorem 3.6.4 (Krajíček, Pudlák (KP90)) $S_2^1 \vdash \mathrm{RFN}(EF)$.

Proof. We have to show

$$S_2^1 \ \vdash \ (\forall\pi)(\forall\varphi)\mathrm{Prf}_{EF}(\pi,\varphi) \to \mathrm{Taut}(\varphi) \ .$$

Assume that $\pi = (\varphi_1, \varphi_2, \ldots, \varphi_n = \varphi)$ is an EF-proof of φ and $S_2^1 \vdash \mathrm{Prf}_{EF}(\pi,\varphi)$. We have to show $S_2^1 \vdash \mathrm{Taut}(\varphi)$ which is by definition

$$S_2^1 \ \vdash \ (\forall\alpha)Assign(\alpha,\varphi) \to \alpha \models \varphi \ .$$

Assume that in the proof π the propositional variables $\bar{p}$ occur together with the extension variables $\bar{q}$. Consider the formula

$$\theta(\alpha,i) = (\exists\beta)Assign(\beta,\bar{q}) \wedge \alpha \cup \beta \models \bigwedge_{j=1}^{i} \varphi_j$$

expressing that the assignment α can be extended to an assignment to the extension variables $\bar{q}$ that satisfies the first i formulas from the proof π.

Formulas and proofs are coded by numbers using a pairing function which at least doubles the numbers in each application. Therefore the PIND-induction scheme available in S_2^1 enables us to use induction on the numbers coding the proof steps φ_i, i.e. we can argue by induction on the number of steps. Hence by verifying the correctness of the EF-axioms and rules in S_2^1 we can prove the formula $\theta(\alpha,n)$ by induction on i in $\theta(\alpha,i)$. Because the extension variables do not occur in $\varphi_n = \varphi$ we have shown

$$\alpha \models \varphi \ .$$

As this was shown for all assignments α we obtain $\mathrm{Taut}(\varphi)$. □

In order to generalize this theorem to the extensions of EF we need the following lemma:

Lemma 3.6.5 *Let $\varphi(x)$ be a Π_1^b-formula. Then*

$$S_2^1 \ \vdash \ (\forall x)\varphi(x) \to (\forall y)\mathrm{Taut}(\|\varphi(x)\|^{|y|}) \ .$$

Proof. The lemma can be proved by induction on the logical complexity of φ. However, we can also derive it from the results proved so far. Namely, let $\varphi(x)$ be a Π^b_1-formula such that

$$S^1_2 \vdash (\forall x)\varphi(x) \ .$$

As the proof of Theorem 3.6.1 formalizes in the theory S^1_2 we get

$$S^1_2 \vdash (\forall y)(\exists \pi)\mathrm{Prf}_{EF}(\pi, \|\varphi(x)\|^{|y|}) \ .$$

Using Theorem 3.6.4 we obtain

$$S^1_2 \vdash (\forall x)\varphi(x) \rightarrow (\forall y)\mathrm{Taut}(\|\varphi(x)\|^{|y|})$$

as claimed. □

Examining the proof of Theorem 3.6.4 again for the extensions $EF + \|\Phi\|$ we get:

Corollary 3.6.6 *Let Φ be a polynomial time decidable set of true Π^b_1-formulas. Then $S^1_2 + \Phi \vdash \mathrm{RFN}(EF + \|\Phi\|)$.*

Proof. The proof proceeds again by induction on i in the formula $\theta(\alpha, i)$ defined in the proof of Theorem 3.6.4. The only difference is that in the induction step for the case that φ_i is a formula of the form $\|\psi\|^n$ with $\psi \in \Phi$ we use the formula ψ which is available as an axiom in $S^1_2 + \Phi$ to derive $\mathrm{Taut}(\|\psi\|^n)$ by Lemma 3.6.3. This suffices to prove $\theta(\alpha, i)$. □

To check property 6 for S^1_2 and EF it remains to show that S^1_2 cannot prove the consistency of any proof system stronger than EF. This is stated in the next theorem.

Theorem 3.6.7 (Krajíček, Pudlák (KP90)) *Let P be a propositional proof system such that*

$$S^1_2 \vdash \mathrm{RFN}(P) \ .$$

Then EF p-simulates P.

As before we state the general result for extensions of EF. We postpone the proof to the next section.

Theorem 3.6.8 *Let Φ be a polynomial time decidable set of true Π^b_1-formulas and let P be a propositional proof system such that*

$$S^1_2 + \Phi \vdash \mathrm{RFN}(P) \ .$$

Then $EF + \|\Phi\|$ p-simulates P.

Combining the Corollaries 3.6.2 and 3.6.6 and Theorem 3.6.8 we obtain

Theorem 3.6.9 *Let Φ be a polynomial time decidable set of true Π^b_1-formulas. Then the proof system $EF + \|\Phi\|$ is strongly regular and corresponds to the theory $S^1_2 + \Phi$. In particular, the system $EF + \|\Phi\|$ has the strong reflection property.*

3.7 Regular Proof Systems

Using the results from Buss (Bus86) and Krajíček and Pudlák (KP90) which we explained in the previous section we will now exhibit sufficient conditions for the regularity of a propositional proof system. From the definition of a regular system as given in Sect. 3.5 it is clear that regular proof systems have the reflection property. Furthermore, a combination of the properties of proof systems introduced in Sect. 2.6 guarantees the regularity of the system, namely:

Theorem 3.7.1 *1. Let P be a proof system such that $EF \leq P$ and P has the reflection property and is closed under substitutions and multiple applications of modus ponens. Then P is regular and corresponds to the theory $S^1_2 + \mathrm{RFN}(P)$. In particular we have*

$$EF + \|\mathrm{RFN}(P)\| \equiv P \ .$$

2. If P is a proof system such that $EF \leq_p P$ and P has the strong reflection property and is efficiently closed under substitutions and multiple applications of modus ponens, then P is strongly regular and corresponds to the theory $S^1_2 + \mathrm{RFN}(P)$. In particular we have

$$EF + \|\mathrm{RFN}(P)\| \equiv_p P \ .$$

The proof of Theorem 3.7.1 requires a series of lemmas which will also be useful in later sections.

Lemma 3.7.2 *Let P be a proof system such that $EF \leq P$ and P is closed under substitutions and multiple applications of modus ponens. Let Φ be some polynomial time set of tautologies such that $P \vdash_* \Phi$. Then*

$$EF + \Phi \leq P \ .$$

Proof. Let $EF + \Phi \vdash_{\leq m} \varphi$. This means that there are substitution instances $\psi_1, \ldots, \psi_k$ of formulas from Φ such that

$$EF \cup \{\psi_1, \ldots, \psi_k\} \vdash_{\leq m} \varphi \ .$$

Using the deduction theorem for EF we get

$$EF \vdash_{\leq p(m)} (\bigwedge_{i=1}^{k} \psi_i) \to \varphi$$

where p is the polynomial from the deduction theorem. By induction on k it can be shown that

$$EF \vdash_{\leq p'(m)} (\psi_1 \to (\psi_2 \to \ldots \to (\psi_k \to \varphi) \ldots)$$

with some polynomial p'. The hypothesis $P \geq EF$ gives us

$$P \vdash_{\leq p''(m)} (\psi_1 \to (\psi_2 \to \ldots \to (\psi_k \to \varphi) \ldots)$$

for some polynomial p''. Since $P \vdash_* \Phi$ and P is closed under substitutions we get polynomial size P-proofs of ψ_i for $i = 1, \ldots, k$. Finally using the closure of P under multiple applications of modus ponens we obtain polynomial size P-proofs of φ. □

Making stronger assumptions we can improve the simulation of $EF + \Phi$ by P from the last lemma to a p-simulation.

Lemma 3.7.3 *Let P be a proof system such that $EF \leq_p P$ and P is efficiently closed under substitutions and multiple applications of modus ponens. Let Φ be some polynomial time set of tautologies such that P-proofs of all formulas from Φ can be constructed in polynomial time. Then*

$$EF + \Phi \leq_p P \ .$$

Proof. As also the deduction property of EF holds in an efficient version (Theorem 2.4.2) the assumptions guarantee that all steps in the proof of Lemma 3.7.2 can be efficiently executed. □

We will mostly use Lemmas 3.7.2 and 3.7.3 in the following form:

Corollary 3.7.4 *1. Let P be a proof system with the reflection property such that $EF \leq P$ and P is closed under substitutions and multiple applications of modus ponens. Then*

$$EF + \|\mathrm{RFN}(P)\| \leq P \ .$$

2. If the proof system $P \geq_p EF$ has the strong reflection property and P is efficiently closed under under substitutions and multiple applications of modus ponens, then we get the p-simulation

$$EF + \|\mathrm{RFN}(P)\| \leq_p P \ .$$

Further comparing the proof systems $EF + \|\mathrm{RFN}(P)\|$ and P we now come to the reverse reduction shown in (KP89). This reduction is even a $\leq_p$-reduction and no assumptions on P are necessary.

Proposition 3.7.5 (Krajíček, Pudlák (KP89)) *Let P be a proof system. Then*

$$P \leq_p EF + \|\mathrm{RFN}(P)\| \ .$$

Proof. Let π be a P-proof of φ. Because $\mathrm{RFN}(P)$ is available as an axiom we get by substitution a polynomial size $EF + \|\mathrm{RFN}(P)\|$-proof of

$$\|\mathrm{Prf}_P(x,y)\|(\bar{p}^x/\bar{\pi}, \bar{p}^y/\bar{\varphi}) \to \|\mathrm{Taut}(y)\|(\bar{p}^y/\bar{\varphi}) \ ,$$

where the suffix $(\bar{p}^x/\bar{\pi})$ indicates that the propositional variables for x are substituted by the bits of π, and similarly for $(\bar{p}^y/\bar{\varphi})$. The formula

$$\|\mathrm{Prf}_P(x,y)\|(\bar{p}^x/\bar{\pi}, \bar{p}^y/\bar{\varphi})$$

can be evaluated in EF to $\top$, giving a polynomial size proof of

$$\|\mathrm{Taut}(y)\|(\bar{p}^y/\bar{\varphi})$$

in the proof system $EF + \|\mathrm{RFN}(P)\|$. From this we get by Lemma 3.6.3 a polynomial size EF-proof the tautology φ. As these proofs can be constructed in polynomial time we get the $\leq_p$-reduction. □

The previous proposition can be seen as a propositional version of property 2 of the correspondence to arithmetic theories and documents the importance of the proof systems $EF + \|\mathrm{RFN}(P)\|$.

For later use we now prove a lemma which is very similar to Proposition 3.7.5.

Lemma 3.7.6 *Let P be a proof system and Φ be some polynomial time set of tautologies. Then*

$$EF + \Phi \vdash_* \|\mathrm{RFN}(P)\|^n \quad \textit{implies} \quad P \leq EF + \Phi \ .$$

Proof. Let π be a P-proof of φ. Because $EF + \Phi \vdash_* \|\mathrm{RFN}(P)\|^n$ and $EF + \Phi$ is closed under substitutions we get a polynomial size $EF + \Phi$-proof of

$$\|\mathrm{Prf}_P(x,y)\|(\bar{p}^x/\bar{\pi}, \bar{p}^y/\bar{\varphi}) \to \|\mathrm{Taut}(y)\|(\bar{p}^y/\bar{\varphi}) \ .$$

$\|\mathrm{Prf}_P(x,y)\|(\bar{p}^x/\bar{\pi}, \bar{p}^y/\bar{\varphi})$ can be evaluated in EF to $\top$, giving a polynomial size $EF + \Phi$-proof of $\|\mathrm{Taut}(y)\|(\bar{p}^y/\bar{\varphi})$. From this we get again by

Lemma 3.6.3 a polynomial size EF-proof of the tautology φ. Combining these proofs by modus ponens we get the $EF + \Phi$-proof of φ. □

Note that the reduction in the last lemma is only $\leq$ as the $EF + \Phi$-proofs of $\|\mathrm{RFN}(P)\|^n$ are not assumed to be constructible in polynomial time. However, if we make this assumption we can draw the stronger conclusion $P \leq_p EF + \Phi$:

Lemma 3.7.7 *Let P be a proof system and Φ be some polynomial time set of tautologies. If $EF + \Phi$-proofs of $\|\mathrm{RFN}(P)\|^n$ can be generated in polynomial time, then $P \leq_p EF + \Phi$.*

Proof. Given a P-proof π of a formula φ we start by generating the $EF + \Phi$-proof of $\|\mathrm{RFN}(P)\|^{|\pi|,|\varphi|}$. Careful analysis of the proof of Lemma 3.7.6 then shows that all transformations can be efficiently performed. Therefore we get the $\leq_p$-simulation. □

Lemma 3.7.7 enables us to give an easy proof of Theorem 3.6.8 from Sect. 3.6.

Theorem 3.6.8 *Let Φ be a polynomial time decidable set of true Π^b_1-formulas and let P be a propositional proof system such that*

$$S^1_2 + \Phi \vdash \mathrm{RFN}(P) \ .$$

Then $EF + \|\Phi\|$ p-simulates P.

Proof. Let P be a proof system such that

$$S^1_2 + \Phi \vdash \mathrm{RFN}(P) \ .$$

As $\mathrm{RFN}(P)$ is a $\forall\Pi^b_1$-formula we conclude with Corollary 3.6.2

$$EF + \Phi \vdash_* \|\mathrm{RFN}(P)\| \ .$$

As these proofs can be constructed in polynomial time we get by Lemma 3.7.7 the simulation $P \leq_p EF + \|\Phi\|$. □

Now we come to the proof of Theorem 3.7.1.

Proof of Theorem 3.7.1. To prove part 1 of the theorem let P be a proof system such that $EF \leq P$ and P has reflection and is closed under substitutions and multiple applications of modus ponens. By Corollary 3.7.4 we have

$$EF + \|\mathrm{RFN}(P)\| \leq P$$

and Proposition 3.7.5 gives

$$P \leq_p EF + \|\mathrm{RFN}(P)\| \ .$$

Hence $EF + \|\mathrm{RFN}(P)\|$ and P are $\leq$-equivalent.

Next we have to check the axioms of the correspondence for $S^1_2 + \mathrm{RFN}(P)$ and P. Suppose φ is a $\forall\Pi^b_1$-formula such that

$$S^1_2 + \mathrm{RFN}(P) \ \vdash \ \varphi \ .$$

By Corollary 3.6.2 we get

$$EF + \|\mathrm{RFN}(P)\| \ \vdash_* \ \|\varphi\|^n \ .$$

As we already know that $EF + \|\mathrm{RFN}(P)\|$ is simulated by P we obtain

$$P \vdash_* \|\varphi\|^n \ .$$

This proves part 1 of the correspondence.

It remains to check the second part. Clearly

$$S^1_2 + \mathrm{RFN}(P) \ \vdash \ \mathrm{RFN}(P) \ .$$

Finally suppose

$$S^1_2 + \mathrm{RFN}(P) \ \vdash \ \mathrm{RFN}(Q)$$

for some proof system Q. By Corollary 3.6.2 this implies

$$EF + \|\mathrm{RFN}(P)\| \vdash_* \|\mathrm{RFN}(Q)\| \ .$$

Now we can apply Lemma 3.7.6 and Corollary 3.7.4 to conclude

$$Q \leq EF + \|\mathrm{RFN}(P)\| \leq P \ .$$

We now prove the second part of the theorem stating that all transformations carried out in the first part are actually polynomial time computable under the stronger assumptions of part 2 of the theorem.

For this let P be a proof system such that $EF \leq_p P$ and P has strong reflection and is efficiently closed under substitutions and multiple applications of modus ponens. By Corollary 3.7.4 we have

$$EF + \|\mathrm{RFN}(P)\| \leq_p P$$

and Proposition 3.7.5 gives

$$P \leq_p EF + \|\mathrm{RFN}(P)\| \ .$$

Hence $EF + \|\mathrm{RFN}(P)\|$ and P are $\leq_p$-equivalent.

We proceed by checking the axioms of strong regularity for $S^1_2 + \mathrm{RFN}(P)$ and P. Suppose φ is a $\forall\Pi^b_1$-formula such that

$$S^1_2 + \mathrm{RFN}(P) \vdash \varphi \ .$$

By Corollary 3.6.2 we can construct $EF + \|\mathrm{RFN}(P)\|$-proofs of $\|\varphi\|^n$ in polynomial time. Because $EF + \|\mathrm{RFN}(P)\| \leq_p P$ we can efficiently translate these $EF + \|\mathrm{RFN}(P)\|$-proofs into P-proofs. This proves part 5 of the correspondence.

For axiom 6 let us assume that

$$S^1_2 + \mathrm{RFN}(P) \vdash \mathrm{RFN}(Q)$$

for a proof system Q. By Corollary 3.6.2 we can construct $EF + \|\mathrm{RFN}(P)\|$-proofs of $\|\mathrm{RFN}(Q)\|^n$ in polynomial time. Now we can apply Lemma 3.7.7 and Corollary 3.7.4 to conclude

$$Q \leq_p EF + \|\mathrm{RFN}(P)\| \leq_p P \ .$$

□

In (Kra95) a sequence of tautologies φ_n is called *hard for a proof system* P if φ_n is constructible in polynomial time, i.e. there exists a polynomial time computable function that produces φ_n on input 1^n, and $P \nvdash_* \varphi_n$. The next theorem from (Kra95) collects some of the most important information on optimal proof systems.

Theorem 3.7.8 (Krajíček (Kra95)) *For all proof systems $P \geq EF$ that are closed under substitutions and multiple applications of modus ponens the following conditions are equivalent:*

1. *There exists a sequence of tautologies hard for P.*
2. *The proof system P is not optimal.*
3. *There is a proof system Q such that $P \nvdash_* \|\mathrm{RFN}(Q)\|^n$.*

Proof. To prove the implication $1 \Rightarrow 2$ let φ_n be a sequence of hard tautologies for P. Consider the proof system $Q = EF + \{\varphi_n \mid n \geq 0\}$. As $P \nvdash_* \varphi_n$ and $Q \vdash_* \varphi_n$ we have $P \not\geq Q$, hence P is not optimal.

For the implication $2 \Rightarrow 3$ let P be a non-optimal proof system. Hence there exists a proof system Q such that $Q \not\leq P$. Then $\|\mathrm{RFN}(Q)\|^n$ is a sequence of tautologies hard for P. Assume on the contrary that $P \vdash_*$

$\|\mathrm{RFN}(Q)\|^n$. Since $P \geq EF$ is closed under substitutions and multiple applications of modus ponens we get by Lemma 3.7.2 and Proposition 3.7.5

$$P \geq EF + \|\mathrm{RFN(Q)}\| \geq Q$$

contradicting $Q \not\leq P$.

As $3 \Rightarrow 1$ is trivial the proof is complete. □

3.8 Comparing Properties of Proof Systems

The observations from the last section allow us to compare some of the properties of propositional proof systems that we introduced in Sect. 2.6. In particular we want to know whether these properties are independent from each other. We will start with the comparison of closure under substitutions and closure under modus ponens.

Proposition 3.8.1 *Assume that the extended Frege proof system is not optimal. Then there exist proof systems which are closed under substitutions but not under modus ponens.*

Proof. We use the assumption that EF is not optimal to get by Theorem 3.7.8 a polynomial time constructible sequence of tautologies ψ_n with $EF \nvdash_* \psi_n$. We may assume that the formulas ψ_n do not contain implications.

Let φ_n be an arbitrary polynomial time constructible sequence of tautologies with polynomially long EF-proofs. We define the system Q as:

$$Q(\pi) = \begin{cases} \varphi & \text{if } \pi = 0\pi' \text{ and } \pi' \text{ is an } EF\text{-proof of } \varphi \\ \sigma(\varphi_n \to \psi_n) & \text{if } \pi = 10^n 1\sigma \text{ for some substitution } \sigma \\ \top & \text{otherwise.} \end{cases}$$

Because EF is closed under substitutions this is also true for Q according to the second line of its definition. From $EF \vdash_* \varphi_n$ and $EF \leq_p Q$ we get $Q \vdash_* \varphi_n$. We also have $Q \vdash_* \varphi_n \to \psi_n$ according to the definition of Q. By hypothesis we have $EF \nvdash_* \psi_n$. Substitution instances of $\varphi_n \to \psi_n$ are different from the formulas ψ_n because the former are implications whereas the latter do not contain the connective $\to$. Therefore also $Q \nvdash_* \psi_n$ and hence Q is not closed under modus ponens. □

Candidates for proof systems that are closed under modus ponens but not under substitutions by constants come from the extensions $EF \cup \Phi$ of EF by polynomial time computable sets $\Phi \subseteq$ TAUT as new axioms. Clearly

the systems $EF \cup \Phi$ are closed under modus ponens. In Sect. 4.13 (Theorem 4.13.10), however, we will exhibit a suitable hypothesis that guarantees that $EF \cup \Phi$ is not closed under substitutions by constants for a suitable choice of Φ.

Full independence of all properties from Sect. 2.6 is not available as the next proposition demonstrates:

Proposition 3.8.2 *Let P be a proof system such that $EF \leq P$ and P has reflection and is closed under substitutions and multiple applications of modus ponens. Then P is also closed under conjunctions and disjunctions.*

Proof. The assumptions guarantee that $P \equiv EF + \|\mathrm{RFN}(P)\|$ by Theorem 3.7.1. The latter system is closed under conjunctions and disjunctions. Because these closure properties are maintained inside a $\leq$-degree they are shared by the system P. □

Most of the properties that we investigated in Sect. 2.6 are robust in the sense that they are preserved inside a $\leq$-or $\leq_p$-degree. As we have seen in this chapter that the reflection property is of central importance for strong systems it is natural to ask whether also the reflection property is robust. The next proposition shows that this is indeed a delicate question as the reflection property of a proof system P even depends on the choice of the Turing machines which are used to evaluate the P-proofs (cf. Sect. 3.4).

Proposition 3.8.3 *Assume that the extended Frege proof system is not p-optimal. Then there exists a proof system $Q \equiv_p EF$ such that*

$$S_2^1 \nvdash_* (\forall\pi)(\forall\varphi)\mathrm{Prf}_Q(\pi,\varphi) \to \mathrm{Taut}(\varphi)$$

for some suitable choice of the Turing machine that computes Q and is used for the formula Prf_Q.

Proof. If EF is not p-optimal, then there exists a proof system R such that $R \not\leq_p EF$. We define the system P as $EF + \|\mathrm{RFN}(R)\|$. By Proposition 3.7.5 we have $R \leq_p P$ and therefore also $P \not\leq_p EF$. We now define the system Q as

$$Q(\pi) = \begin{cases} \varphi & \text{if } \pi = 0\pi' \text{ and } \pi' \text{ is an } EF\text{-proof of } \varphi \\ P(\pi') & \text{if } \pi = 1\pi' \text{ and } P(\pi') \in \{\top, \bot\} \\ \top & \text{otherwise.} \end{cases}$$

Then EF and Q are $\leq_p$-equivalent because EF $\leq_p$-reduces to Q via $\pi \mapsto 0\pi$ and the opposite reduction $Q \leq_p EF$ is given by

$$\pi \mapsto \begin{cases} \pi' & \text{if } \pi = 0\pi' \\ \pi_0 & \text{if } \pi = 1\pi' \end{cases}$$

where π_0 is a fixed EF-proof of $\top$. We have to show that S_2^1 does not prove the formula $\mathrm{RFN}(Q)$ where for the predicate Prf_Q we use the canonical Turing machine M according to the above definition of Q, i.e. on input $0\pi'$ the machine M checks whether π' is a correct EF-proof and on input $1\pi'$ the machine M evaluates $P(\pi')$. Assume on the contrary that $S_2^1 \vdash_* \mathrm{RFN}(Q)$. Because of line 2 of the definition of Q this means that S_2^1 can prove that there is no P-proof of $\bot$, i.e. S_2^1 proves the consistency statement of P. The system P is closed under substitutions by constants and modus ponens. Therefore $\mathrm{Con}(P)$ and $\mathrm{RFN}(P)$ are equivalent in S_2^1 by Proposition 3.4.1. Together with $S_2^1 \vdash \mathrm{Con}(P)$ this yields $S_2^1 \vdash \mathrm{RFN}(P)$, and hence by Theorem 3.6.7 we obtain $P \leq_p EF$, contradicting the choice of P. Thus S_2^1 proves $\mathrm{RFN}(EF)$ but not $\mathrm{RFN}(Q)$. □

Chapter 4

Disjoint NP-Pairs

Daher ist das schönste Zeichen der Originalität, wenn man einen empfangenen Gedanken dergestalt fruchtbar zu entwickeln weiß, daß niemand leicht, wie viel in ihm verborgen liege, gefunden hätte.

Johann Wolfgang Goethe

This chapter is devoted to the study of disjoint NP-pairs. We start with a complexity theoretic analysis of the basic definitions and some observations about the simulation order of disjoint NP-pairs. Our main objective, however, is to explore the close connection between NP-pairs and propositional proof systems. In particular, this also involves the correspondence to bounded arithmetic as developed in the previous chapter.

4.1 Reductions Between NP-Pairs

Definition 4.1.1 *A pair (A, B) is called a* disjoint NP-pair (DNPP) *if the components A and B are in* NP *and $A \cap B = \emptyset$. To exclude trivial cases we additionally require $A \neq \emptyset$ and $B \neq \emptyset$.*

The set of all disjoint NP-pairs can be considered as a promise complexity class, denoted by DisjNP in (GSSZ04) and subsequent papers by these authors. The machine model consists of pairs (M_1, M_2) of nondeterministic polynomial time Turing machines with the promise that there does not exist any input that is accepted by both machines M_1 and M_2.

The complexity theoretic investigation of disjoint NP-pairs began with the work of Even, Selman and Yacobi (ESY84) and Grollmann and Selman (GS88). Their main motivation was to provide a complexity theoretic

framework for the analysis of the security of public-key crypto systems. Security aspects of a public-key cryptosystem can then be modeled by a disjoint NP-pair associated with the crypto system. We will briefly describe this application of disjoint NP-pairs in Sect. 5.1.

An important concept in the work of Grollmann and Selman (GS88) is the notion of a separator of a disjoint NP-pair, defined as follows:

Definition 4.1.2 *A set S is a* separator *for the disjoint* NP*-pair (A, B) if $A \subseteq S$ and $B \subseteq \overline{S}$.*

Of central interest is the case where a given DNPP has a separator which is computable in polynomial time. If this is the case, then the pair is called *p-separable*, otherwise *p-inseparable*.

Formulated differently, a disjoint NP-pair (A, B) is p-separable if there exists a polynomial time computable function f that outputs 1 on inputs from A and 0 on inputs from B and answers arbitrarily otherwise. This makes it clear that disjoint NP-pairs are indeed promise problems (Gol05).

Whether or not all disjoint NP-pairs are p-separable is an open problem. Concrete candidates for p-inseparable pairs are provided by cryptographic pairs (cf. Sect. 5.1) and pairs defined from propositional proof systems (cf. Sect. 4.4). It is known that p-inseparable pairs exist under suitable assumptions. For example $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$ is such an assumption: take a set $A \in (\mathsf{N}P \cap coNP) \setminus P$. Then $(A, \bar{A})$ is a p-inseparable disjoint NP-pair. Grollmann and Selman (GS88) showed that also $\mathsf{P} \neq \mathsf{UP}$ is a sufficient condition for the existence of p-inseparable DNPP. However, it is not known how to derive the existence of p-inseparable disjoint NP-pairs from the assumption $\mathsf{P} \neq \mathsf{NP}$. Homer and Selman (HS92) also constructed an oracle relative to which $\mathsf{P} \neq \mathsf{NP}$ but p-inseparable disjoint NP-pairs do not exist. Therefore the existence of p-inseparable DNPP is a condition which currently appears to be in strength intermediate between $\mathsf{P} \neq \mathsf{NP}$ and $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$.

If we aim to consider DisjNP as a complexity class we need reductions which are suitable for pairs. Grollmann and Selman defined in (GS88) a variety of these, the most common being the following kind of the many-one reduction:

Definition 4.1.3 (Grollmann, Selman (GS88)) *A pair (A, B) is* polynomially reducible *to a DNPP (C, D), denoted by $(A, B) \leq_p (C, D)$, if there exists a polynomial time computable function f such that $f(A) \subseteq C$ and $f(B) \subseteq D$.*

As usual we define an equivalence relation $\equiv_p$ as: $(A, B) \equiv_p (C, D)$ if $(A, B) \leq_p (C, D)$ and $(C, D) \leq_p (A, B)$. The equivalence classes of $\equiv_p$ are called degrees.

The reason why $\leq_p$ should be considered as a many-one reduction lies in the following non-uniform characterization of $\leq_p$, shown in (GSSZ04):

Theorem 4.1.4 (Glaßer, Selman, Sengupta, Zhang (GSSZ04)) *Let (A, B) and (C, D) be disjoint* NP*-pairs. Then $(A, B) \leq_p (C, D)$ if and only if for every separator T of (C, D) there exists a separator S of (A, B) such that $S \leq_m^p T$.*

The characterization of $\leq_p$ in Theorem 4.1.4 is a natural notion of a reduction in the context of promise problems (cf. (Gol05)). It expresses that for every separator T of (C, D) there is a separator of (A, B) that is not more complex than T. However, the uniform version of this reduction as given in Definition 4.1.3 is much easier to work with.

If f performs a $\leq_p$-reduction from (A, B) to (C, D), then f is also allowed to map elements from the complement of $A \cup B$ to C or D. Therefore $f : (A, B) \leq_p (C, D)$ does not imply in general that f is a many-one reduction between A and C or between B and D. This, however, is the case for the following stronger reduction:

Definition 4.1.5 (Köbler, Messner, Torán (KMT03)) *A disjoint* NP*-pair (A, B) is* strongly reducible *to a DNPP (C, D), denoted by $(A, B) \leq_s (C, D)$, if there exists a polynomial time computable function f such that $f^{-1}(C) = A$ and $f^{-1}(D) = B$.*

Analogously to $\equiv_p$ we define $\equiv_s$ as the equivalence relation associated with $\leq_s$.

Equivalently, we can view $\leq_s$ as a reduction between triples. In addition to the two conditions $f(A) \subseteq C$ and $f(B) \subseteq D$ for $\leq_p$ we also require $f(\overline{A \cup B}) \subseteq \overline{C \cup D}$.

The reduction $\leq_s$ now has the property that if f realizes a $\leq_s$-reduction from (A, B) to (C, D), then f is simultaneously a many-one-reduction between A and C as well as between B and D. Clearly, this also serves as a characterization of $\leq_s$, namely:

Proposition 4.1.6 *Let (A, B) and (C, D) be DNPP. Then $(A, B) \leq_s (C, D)$ if and only if there exists a function $f \in$ FP such that $f : A \leq_m^p C$ and $f : B \leq_m^p D$.*

In contrast the reduction $\leq_p$ is more flexible because here a reduction $f : (A, B) \leq_p (C, D)$ does not relate the complexity of A or B to the complexity of C or D, respectively. We may express this differently as:

Proposition 4.1.7 *For every DNPP (A, B) there exists a DNPP (A', B') such that $(A, B) \equiv_p (A', B')$ and A', B' are* NP*-complete.*

Proof. Choose $A' = A \times \mathrm{SAT}$ and $B' = B \times \mathrm{SAT}$. Then we have $(A, B) \leq_p (A', B')$ via $x \mapsto (x, \varphi_0)$ with a fixed formula $\varphi_0 \in \mathrm{SAT}$, and $(A', B') \leq_p (A, B)$ via the projection $(x, \varphi) \mapsto x$. □

Obviously $\leq_s$ is a refinement of $\leq_p$. It is indeed a proper refinement. For this let (C, D) be a disjoint NP-pair that has an empty complement $\overline{C \cup D}$. Let (A, B) be a second DNPP such that $(A, B) \leq_p (C, D)$ but with nonempty complement $\overline{A \cup B}$. Then it is obviously not possible to map (A, B) to (C, D) with the stronger reduction $\leq_s$ as there are no elements for the image of $A \cup B$. Examples for such pairs (A, B) and (C, D) are easy to find. Nevertheless this separation is not very satisfactory as it only applies to pairs with empty complement. But it is possible to achieve a separation which, although conditional, only involves pairs where all three components are nonempty. This separation was first observed in (GSS05). Using Proposition 4.1.7 we can give an easy proof.

Proposition 4.1.8 (Glaßer, Selman, Sengupta (GSS05))
There exist disjoint NP*-pairs* (A, B) *and* (C, D) *such that* $\overline{A \cup B}$ *and* $\overline{C \cup D}$ *are nonempty and* $(A, B) \leq_p (C, D)$*, but* $(A, B) \not\leq_s (C, D)$ *if and only if* $\mathsf{P} \neq \mathsf{NP}$*.*

Proof. Let C and D be nonempty disjoint sets in P such that $\overline{C \cup D}$ is also nonempty. By Proposition 4.1.7 there exist NP-complete sets A and B such that $(A, B) \equiv_p (C, D)$. But (A, B) is not strongly reducible to (C, D) because $(A, B) \leq_s (C, D)$ would imply in particular $A \leq_m^p C$ and hence $\mathsf{P} = \mathsf{NP}$.

On the other hand if $\mathsf{P} = \mathsf{NP}$, then all DNPP (A, B) where all three components $A, B, \overline{A \cup B}$ are nonempty are $\leq_s$-equivalent. □

Alternatively, Proposition 4.1.8 can be formulated for pairs (A, B) and (C, D) with infinite components A,B, C, D, $\overline{A \cup B}$ and $\overline{C \cup D}$.

Apart from the many-one reductions $\leq_p$ and $\leq_s$ there is also a natural notion of Turing reduction for pairs, defined already by Grollmann and Selman:

Definition 4.1.9 (Grollmann, Selman (GS88)) *Let* (A, B) *and* (C, D) *be DNPP. We say that* (A, B) *is* Turing reducible *to* (C, D)*, denoted by* $(A, B) \leq_T (C, D)$*, if there exists a polynomial time oracle Turing machine* M *such that for every separator* T *of* (C, D) $L(M^T)$ *separates* (A, B)*.*

If for inputs from $A \cup B$ *the machine* M *makes only queries to* $C \cup D$ *we call the reduction performed by* M *a* smart *Turing reduction.*

Again this uniform formulation of a Turing reduction has an equivalent non-uniform counterpart:

Theorem 4.1.10 (Grollmann, Selman (GS88)) *Let (A, B) and (C, D) be disjoint* NP*-pairs. Then $(A, B) \leq_T (C, D)$ if and only if for every separator T of (C, D) there exists a separator S of (A, B) such that $S \leq_T^p T$.*

In this characterization the Turing reduction $S \leq_T^p T$ may be performed by different Turing machines for different separators T, whereas in Definition 4.1.9 there is one fixed machine M that defines a separator $S = L(M^T)$ of (A, B) for all separators T of (C, D).

4.2 Some Remarks on the Simulation Order of Disjoint NP-Pairs

The set of all p-separable DNPP forms the minimal degree with respect to the $\leq_p$-reduction, namely:

Proposition 4.2.1 *Let (A, B) be a p-separable DNPP. Then (A, B) is $\leq_p$-reducible to any other disjoint* NP*-pair. If on the other hand a pair (C, D) is $\leq_p$-reducible to (A, B) then also (C, D) is p-separable.*

Proof. Let (A, B) be p-separable and let $S \in \mathsf{P}$ be a separator of (A, B). If (C, D) is an arbitrary disjoint NP-pair and $c_0 \in C$ and $d_0 \in D$ are fixed elements from its components, then

$$x \mapsto \begin{cases} c_0 & \text{if } x \in S \\ d_0 & \text{otherwise} \end{cases}$$

is a $\leq_p$-reduction from (A, B) to (C, D).

Assume now that the pair (A, B) is separated by the function $f \in \mathsf{FP}$, i.e.

$$\begin{aligned} x \in A \quad &\Longrightarrow \quad f(x) = 1 \\ x \in B \quad &\Longrightarrow \quad f(x) = 0 \ . \end{aligned}$$

If (C, D) is a disjoint NP-pair that is $\leq_p$-reducible to(A, B) via the polynomial time computable reduction g, then $f \circ g$ separates the pair (C, D). □

It is clear that this proposition is also valid for Turing reductions:

Proposition 4.2.2 *The minimal $\leq_T$-degree consists of all p-separable* NP*-pairs.*

Proof. Proposition 4.2.1 implies that every p-separable pair is $\leq_T$-reducible to any other pair. If on the other hand (C, D) is p-separable by a separator $T \in \mathsf{P}$ and $(A, B) \leq_T (C, D)$, then by definition there exists a separator S of (A, B) such that $S \leq_T^p T$. Hence $S \in \mathsf{P}$ and (A, B) is also p-separable. □

For the stronger $\leq_s$-reduction this minimal degree shrinks to the set of all p-separable pairs with empty complement, i.e. sets of the form $(A, \bar{A})$ with $A \in \mathsf{P}$:

Proposition 4.2.3 *Let A be a set in P. Then $(A, \bar{A})$ is $\leq_s$-reducible to any other disjoint NP-pair. If on the other hand a pair (C, D) is $\leq_s$-reducible to $(A, \bar{A})$, then $D = \bar{C}$ and $C \in \mathsf{P}$.*

But assuming $\mathsf{P} \neq \mathsf{NP}$ also the set of all p-separable pairs with nonempty complement splits into different $\equiv_s$-degrees. The precise picture of all p-separable DNPP under $\leq_s$ is given in the next proposition.

Proposition 4.2.4 *If $\mathsf{P} \neq \mathsf{NP}$, then there exist infinitely many distinct $\leq_s$-degrees of p-separable disjoint NP-pairs.*

More precisely, if $(\mathcal{A}, \leq_{\mathsf{NP}})$ is the order of all $\leq_m^p$-degrees of NP-sets, excluding the empty set, then $(0 \cup \mathcal{A} \times \mathcal{A}, \leq_{\mathsf{DNPP}})$ is the order of all $\leq_s$-degrees of p-separable disjoint NP-pairs, where $0 \notin \mathcal{A}$. The relation $\leq_{\mathsf{DNPP}}$ is defined as:

- $0 \leq_{\mathsf{DNPP}} (X, Y)$ *for all* $X, Y \in \mathcal{A}$ *and*
- $(X_1, Y_1) \leq_{\mathsf{DNPP}} (X_2, Y_2)$ *if* $X_1 \leq_{\mathsf{NP}} X_2$ *and* $Y_1 \leq_{\mathsf{NP}} Y_2$.

Further, the minimal $\leq_s$-degree 0 consists of all disjoint NP-pairs of the form $(A, \bar{A})$ with $A \in \mathsf{P}$, and a $\leq_s$-degree $(X, Y) \in \mathcal{A} \times \mathcal{A}$ is equal to

$$\{(A, B) \mid A \cap B = \emptyset,\ A \in X,\ B \in Y,\ \text{and } (A, B) \text{ is p-separable}\}\ .$$

Proof. By a theorem of Ladner (Lad75) there exist infinitely many different $\leq_m^p$-degrees of NP-sets assuming $\mathsf{P} \neq \mathsf{NP}$. Therefore Ladner's theorem together with the following claim imply the proposition.

Claim: *Let (A, B) and (C, D) be p-separable disjoint NP-pairs such that $\overline{C \cup D} \neq \emptyset$. Then $(A, B) \leq_s (C, D)$ if and only if $A \leq_m^p C$ and $B \leq_m^p D$.*

The first direction is clear from the definition of $\leq_s$.

For the reverse implication let $S, T \in \mathsf{P}$ be separators of (A, B) and (C, D), respectively, and let x_0 be a fixed element from $\overline{C \cup D}$. Let further

$g_1 : A \leq_m^p C$ and $g_2 : B \leq_m^p D$ compute the respective many-one reductions. Then the polynomial time computable function

$$x \mapsto \begin{cases} g_1(x) & \text{if } x \in S \text{ and } g_1(x) \in T \\ g_2(x) & \text{if } x \notin S \text{ and } g_2(x) \notin T \\ x_0 & \text{otherwise} \end{cases}$$

is a $\leq_s$-reduction from (A, B) to (C, D). □

Now we know that the minimal $\leq_p$-degree splits into infinitely many $\leq_s$-degrees if $\mathsf{P} \neq \mathsf{NP}$, and into two $\leq_s$-degrees otherwise. This is essentially the separation we proved in Proposition 4.1.8. But is $\leq_s$ also a proper refinement of $\leq_p$ on other $\leq_p$-degrees? This question is hard to answer as even under the assumption $\mathsf{P} \neq \mathsf{NP}$ we do not know whether there exist $\leq_p$-degrees that are different from the minimal one. At least we can make the following remark:

Proposition 4.2.5 *If* $\mathsf{P} \neq \mathsf{NP}$*, then every* $\leq_p$*-degree that contains a disjoint* NP*-pair* (A, B) *such that* A *or* B *are not* NP*-complete splits into infinitely many* $\leq_s$*-degrees.*

Proof. Let (A, B) be a DNPP such that A is not NP-complete. Let C be an NP-set such that $A \leq_m^p C$ but $C \not\leq_m^p A$. Consider the pair

$$(A \times C, B \times C) \ .$$

This pair is $\leq_p$-equivalent to (A, B) because $(A, B) \leq_p (A \times C, B \times C)$ via $x \mapsto (x, c_0)$ with fixed $c_0 \in C$ and $(A \times C, B \times C) \leq_p (A, B)$ via the projection $(x, y) \mapsto x$. But clearly $(A \times C, B \times C) \not\leq_s (A, B)$ because $A \times C \not\leq_m^p A$.

As by Ladner's result (Lad75) there exist infinitely many such C that are pairwise $\leq_m^p$-inequivalent we get the proposition. □

Even assuming the existence of p-inseparable disjoint NP-pairs we do not know if there are also p-inseparable pairs with components which are not NP-complete. But the cryptographic pairs defined in Sect. 5.1 provide candidates for such NP-pairs. Hence we conjecture that $\leq_s$ is indeed an interesting refinement of $\leq_p$ on the whole class of disjoint NP-pairs.

Actually, under the assumption $\mathsf{P} \neq \mathsf{NP}$ the $\leq_p$-degrees do not only split into infinitely many $\leq_s$-degrees but also reductions between pairs from different $\leq_p$-degrees are not necessarily preserved. We illustrate this for the p-separable pairs in the next proposition.

Proposition 4.2.6 *Assume that there exists a p-inseparable disjoint* NP*-pair* (C, D) *such that* C *is not* NP*-complete. Then there exists a pair* (A, B) *that is p-separable, but still* $(A, B) \not\leq_s (C, D)$.

Proof. We choose a p-separable pair (A, B) where both components A and B are NP-complete. Because the existence of p-inseparable pairs implies $\mathsf{P} \neq \mathsf{NP}$ it is impossible to $\leq_s$-reduce (A, B) to (C, D). □

Having good information on minimal degrees it is natural to ask about maximal degrees under the reductions. Keeping with common terminology we call a disjoint NP-pair $\leq_p$-*complete* if every DNPP $\leq_p$-reduces to it. Similarly, we speak of $\leq_s$- and $\leq_T$-complete pairs. Razborov first raised the following question:

Problem 4.2.7 (Razborov (Raz94)) *Do complete disjoint* NP*-pairs exist?*

This question is one of the most important in the field of disjoint NP-pairs and has been intensively studied (Raz94; KMT03; GSSZ04; GSS05; Bey04a). Nevertheless, so far we only have conditional results relating the existence of optimal proof systems and the existence of complete disjoint NP-pairs. We will describe these results in Sect. 4.11.

We continue with some remarks on the degree structure of disjoint NP-pairs.

Proposition 4.2.8 (Pudlák (Pud03)) *The set of degrees of disjoint* NP*-pairs with the order inherited from the reduction* $\leq_p$ *forms a lattice. The supremum of the degrees of two pairs* (A, B) *and* (C, D) *is the degree of*

$$(A, B) \vee (C, D) = (A \dot\cup C, B \dot\cup D)$$

where $\dot\cup$ *is the marked union of two sets* A, C *over* $\{0, 1\}$*, defined by*

$$A \dot\cup C = \{0x \mid \in A\} \cup \{1x \mid x \in C\} \ ,$$

and the infimum is defined by

$$(A, B) \wedge (C, D) = (A \times C, B \times D) \ .$$

Proof. Two DNPP (A, B) and (C, D) reduce to their supremum $(A \dot\cup C, B \dot\cup D)$ via $x \mapsto 0x$ and $x \mapsto 1x$, respectively.

If (E, F) is some disjoint NP-pair such that $f : (A, B) \leq_p (E, F)$ and $g : (C, D) \leq_p (E, F)$, then

$$x \mapsto \begin{cases} f(y) & \text{if } x = 0y \\ g(y) & \text{if } x = 1y \end{cases}$$

is a $\leq_p$-reduction from $(A \dot\cup C, B \dot\cup D)$ to (E, F).

The infimum $(A \times C, B \times D)$ reduces to (A, B) and (C, D) via the projections on the first and second coordinate, respectively.

Finally, if for some pair (E, F) we have $f : (E, F) \leq_p (A, B)$ and $g : (E, F) \leq_p (C, D)$, then $(E, F) \leq_p (A \times C, B \times D)$ via $x \mapsto (f(x), g(x))$. □

Analysing this situation for the stronger reduction $\leq_s$ we remark:

Proposition 4.2.9 *The set of degrees of disjoint* NP*-pairs with the order inherited from the reduction $\leq_s$ forms an upper semi-lattice. As in the case of the weaker reduction $\leq_p$ the supremum of two degrees represented by the pairs (A, B) and (C, D) is the degree of $(A \dot\cup C, B \dot\cup D)$.*

Proof. The proof is analogous to the proof of Proposition 4.2.8. □

Finally, we mention a recent result of Glaßer, Selman, and Zhang, which shows that the degree structure of all DNPP under Turing reductions is dense.

Theorem 4.2.10 (Glaßer, Selman, Zhang (GSZ05))
Let (A, B) and (C, D) be two disjoint NP*-pairs with infinite components A, B, C, and D such that*

$$(A, B) \leq_T (C, D) \quad \textit{but} \quad (C, D) \not\leq_T (A, B) \ .$$

Then there exist incomparable, strictly intermediate disjoint NP*-pairs (E, F) and (G, H) between (A, B) and (C, D) such that E, F, G, and H are infinite. Precisely, the following properties hold:*

1. *$(A, B) \leq_m (E, F) \leq_T (C, D)$ and $(C, D) \not\leq_T (E, F) \not\leq_T (A, B)$;*
2. *$(A, B) \leq_m (G, H) \leq_T (C, D)$ and $(C, D) \not\leq_T (G, H) \not\leq_T (A, B)$;*
3. *$(E, F) \not\leq_T (G, H)$ and $(G, H) \not\leq_T (E, F)$.*

4.3 Combinatorially Defined Pairs

Let us consider a first example for a disjoint NP-pair. The Clique-Colouring pair (CC_0, CC_1) is defined as follows:

$$\begin{aligned} CC_0 &= \{(G, k) \mid G \text{ is a graph containing a clique of size } k\} \\ CC_1 &= \{(G, k) \mid \text{the graph } G \text{ can be coloured by } k-1 \text{ colours}\} \ . \end{aligned}$$

The pair (CC_0, CC_1) is even an example for a p-separable pair as shown by Lovász (Lov79).

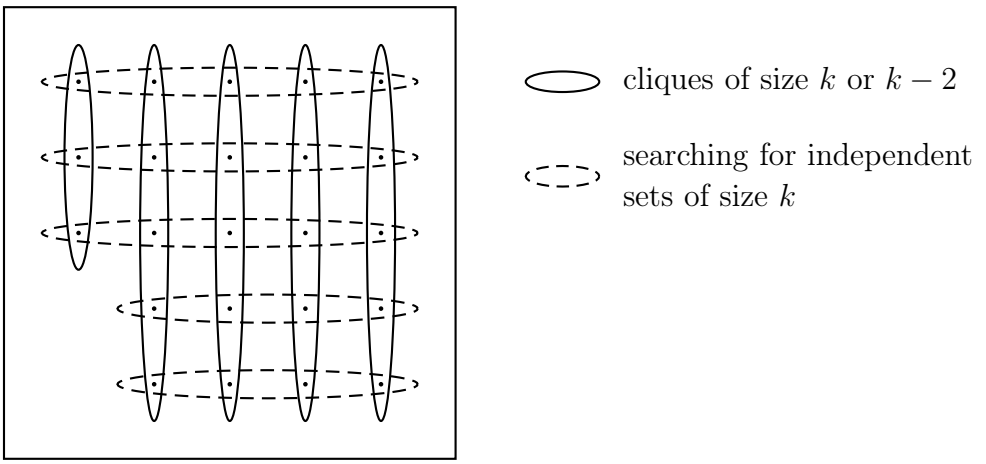

Figure 4.1: The Broken Mosquito Screen

Another combinatorially defined pair is the Broken-Mosquito-Screen pair BMS, introduced by Haken (CH99):

$$\begin{aligned} BMS_0 = \{(G,k) \mid\ & G \text{ has } k^2-2 \text{ vertices and contains } k \text{ disjoint} \\ & \text{cliques of which } k-1 \text{ are of size } k \\ & \text{and 1 is of size } k-2\} \\ BMS_1 = \{(G,k) \mid\ & G \text{ has } k^2-2 \text{ vertices and contains } k \text{ disjoint} \\ & \text{independent sets of which } k-1 \text{ are of size } k \\ & \text{and 1 is of size } k-2\} \end{aligned}$$

The name of the pair becomes apparent from its graphical representation (Fig. 4.1). It is clear that both components are in NP. The disjointness of the pair is also easy to see. Let (G,k) belong to BMS_0. Searching for independent sets of size k in G it is clear that each such set can contain at most one vertex from each clique. But as one clique contains only $k-2$ vertices we can find at most $k-2$ independent sets of size k. Hence $(G,k) \notin BMS_1$. It is not so clear that (BMS_0, BMS_1) is also p-separable. In fact it was even proposed as the basis of a bit commitment scheme. However, Pudlák (Pud03) gave a $\leq_p$-reduction from (BMS_0, BMS_1) to the Clique-Colouring pair, thereby showing that (BMS_0, BMS_1) is p-separable.

Finding meaningful combinatorial disjoint NP-pairs does not seem to be

easy. It is even more complicated to come up with combinatorially defined pairs that could serve as candidates for p-inseparable DNPP. Pudlák (Pud03) discusses some pairs that have not been separated so far, but there is no particular evidence to support the believe in their p-inseparability.

4.4 Disjoint NP-Pairs Characterize Properties of Propositional Proof Systems

In this section we will discuss disjoint NP-pairs which are defined from propositional proof systems. The link between NP-pairs and proof systems was established by Razborov in (Raz94). There he defined a canonical pair from a proof system which corresponds to the reflection property of the system. Pudlák (Pud03) showed that also the automatizability of the proof system and the feasible interpolation property are expressible by disjoint NP-pairs. In this way disjoint NP-pairs have substantially contributed to the understanding of propositional proof systems.

4.4.1 The Canonical Pair of a Proof System

Razborov (Raz94) was the first to associate a disjoint NP-pair $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ with a proof system P. This pair is called the *canonical pair of* P and is defined as follows:

$$\begin{aligned} \mathrm{Ref}(P) &= \{(\varphi, 1^m) \mid P \vdash_{\leq m} \varphi\} \\ \mathrm{SAT}^* &= \{(\varphi, 1^m) \mid \neg\varphi \in \mathrm{SAT}\} \ . \end{aligned}$$

The first component $\mathrm{Ref}(P)$ contains tautologies together with information on their proof length in P, whereas the second component SAT^* is a modified version of SAT that contains all formulas which are not tautological. Clearly, both components are in NP and disjoint. It is also interesting to have a look at the complement of $\mathrm{Ref}(P) \cup \mathrm{SAT}^*$. It contains tautologies together with lower bounds to the proof length of these formulas in the system P, i.e.

$$\overline{\mathrm{Ref}(P) \cup \mathrm{SAT}^*} = \{(\varphi, 1^m) \mid \varphi \in \mathrm{TAUT} \text{ and } P \nvdash_{\leq m} \varphi\} \ .$$

Hence, all information on proof length of P is coded in the canonical pair of P.

Originally, Razborov gave a slightly different definition of the canonical pair. Namely, he considered proof systems as refutation systems that prove formulas by refuting their negations. This is the case for a number of proof

systems like resolution or the cutting planes system. Therefore Razborov formalized the canonical pair as

$$\begin{aligned} \mathrm{Ref}(P) &= \{(\varphi, 1^m) \mid P \vdash_{\leq m} \neg\varphi\} \\ \mathrm{SAT}^* &= \{(\varphi, 1^m) \mid \varphi \in \mathrm{SAT}\} \ . \end{aligned}$$

Refutation systems gave their name to the first component whereas the second component is now simply a padded version of SAT. Because the general definition of Cook and Reckhow (Definition 2.2.1) does not formalize proof systems as refutation systems we decided to modify the content of the canonical pair while keeping its name.

4.4.2 The Canonical Pair and Automatizability

The central question of propositional proof complexity can be stated as follows: given a formula φ and a propositional proof system P, what is the minimal length of a P-proof of φ? However, for many applications it is more important to construct proofs than to merely estimate their length. But we immediately find limitations to the goal of the efficient construction of proofs: constructing proofs in time polynomial in the length of the input formula is not possible unless the system is polynomially bounded. Therefore for the purpose of proof search the notion of efficiency has to be formulated less restrictively, as is done in the next definition:

Definition 4.4.1 (Bonet, Pitassi, Raz (BPR00)) *A proof system P is* automatizable *if there exists a deterministic procedure that takes as input a formula φ and outputs a P-proof of φ in time polynomial in the length of the shortest P-proof of φ if φ is a tautology. If $\varphi \notin$ TAUT, then the behaviour of the algorithm is unspecified.*

For practical purposes automatizable systems would be very desirable. Searching for a proof we may not find the shortest one, but we are guaranteed to find one that is only polynomially longer. Clearly, the truth-table method is automatizable. Apart from this trivial example no other natural proof system is known to be automatizable. Even worse, the information below suggests that finding natural automatizable systems is a too ambitious enterprise.

The automatizability of a proof system P has the following easy characterization in terms of the set $\mathrm{Ref}(P)$:

Proposition 4.4.2 *A proof system P is automatizable if and only if there exists a deterministic polynomial time algorithm that takes as input $(\varphi, 1^m)$*

and produces a P-proof of φ if $(\varphi, 1^m) \in \mathrm{Ref}(P)$. The algorithm might also output proofs for tautologies φ with $(\varphi, 1^m) \notin \mathrm{Ref}(P)$, but in case the proof search fails the algorithm returns some fixed output indicating that no P-proof was found.

Proof. Assume that P is automatizable via the algorithm A. By definition there exists a polynomial p that bounds the running time of A for inputs φ with minimal proof size m in P. Running this algorithm A on input $(\varphi, 1^m)$ for $p(m)$ steps gives a polynomial time algorithm that returns P-proofs for inputs from $\mathrm{Ref}(P)$.

If conversely B is an algorithm for $\mathrm{Ref}(P)$ as specified in the proposition, then the following procedure certifies the automatizability of P:

```
1  Input:  a formula φ
2  m=1
3  REPEAT
4    simulate B(φ, 1^m)
5    IF B(φ, 1^m) returns a P-proof of φ THEN
6      output this P-proof
4    m=m+1
5  UNTIL m=0
```

For tautologies φ with minimal proof size m in P this algorithm executes the REPEAT loop at most m times. Hence in this case the running time is bounded by $O(mp(|\varphi| + m))$ where p is a polynomial for the running time of B. For inputs φ that are not tautological the above algorithm does not terminate. □

From this reformulation of automatizability it is clear that automatizable proof systems have p-separable canonical pairs:

Proposition 4.4.3 (Pudlák (Pud03)) *Let P be an automatizable proof system. Then $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ is p-separable.*

The converse is probably not true as we will show with the following example.

Proposition 4.4.4 *There exists a proof system P that has a p-separable canonical pair. But P is not automatizable unless $\mathsf{P} = \mathsf{NP}$.*

Proof. We define the proof system P as follows:

$$P(\pi) = \begin{cases} \varphi & \text{if } \pi = (\varphi, 1^m) \text{ and } m \geq 2^{|\varphi|} \\ \varphi \vee \top & \text{if } \pi = (\varphi, \alpha) \text{ and } \alpha \text{ is a satisfying assignment for } \varphi \\ \top & \text{otherwise .} \end{cases}$$

The following algorithm separates the canonical pair of P:

```
1  Input:  (φ, 1^m)
2  IF φ = ψ ∨ ⊤ or φ = ⊤ THEN output 1
3  IF m ≥ 2^|φ| THEN
4    IF φ ∈ TAUT THEN output 1
5  output 0 .
```

The test $\varphi \in \mathrm{TAUT}$ in line 4 can be performed in polynomial time by checking all assignments because the parameter m is big enough according to line 3. Hence the algorithm is efficient.

Since formulas $\varphi = \psi \vee \top$ are always tautological the algorithm only outputs 1 if the formula φ is a tautology. Therefore $(\varphi, 1^m) \in \mathrm{SAT}^*$ always leads to the answer 0 whereas inputs $(\varphi, 1^m) \in \mathrm{Ref}(P)$ are always answered by 1 according to lines 2 and 4.

The proof system P is not automatizable because this would mean that on input $\varphi \vee \top$ we would have to produce in polynomial time a satisfying assignment of φ provided $\varphi \in \mathrm{SAT}$. This implies in particular the existence of a deterministic polynomial time algorithm to decide SAT and hence $\mathsf{P} = \mathsf{NP}$. □

This example is not entirely satisfactory as the proof system constructed in the last proof is not very natural. But it might be hard to prove Proposition 4.4.4 for natural proof systems as it is conjectured that the canonical pairs of all studied proof systems are not p-separable (cf. (Pud03)). At least for proof systems stronger than bounded-depth Frege systems we have good reason to believe that their canonical pairs are not p-separable because cryptographic pairs reduce to the canonical pairs of these systems (KP98; BPR00; BDG$^+$04).

As we have seen the p-separability of the canonical pair might not imply the automatizability of the system but at least it implies that there exists a stronger automatizable system as the next theorem by Pudlák shows.

Theorem 4.4.5 (Pudlák (Pud03)) *Let P be a proof system. Then the canonical pair of P is p-separable if and only if there exists an automatizable proof system Q which p-simulates P.*

Proof. Let $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ be separated by the polynomial time computable function f, i.e.

$$\begin{aligned} (\varphi, 1^m) \in \mathrm{Ref(P)} \quad &\Longrightarrow \quad f(\varphi, 1^m) = 1 \\ (\varphi, 1^m) \in \mathrm{SAT}^* \quad &\Longrightarrow \quad f(\varphi, 1^m) = 0 \ . \end{aligned}$$

We define the system Q by

$$Q(\pi) = \begin{cases} \varphi & \text{if } \pi = (\varphi, 1^m) \text{ and } f(\varphi, 1^m) = 1 \\ \top & \text{otherwise .} \end{cases}$$

Let us first check that Q is a proof system. If $f(\varphi, 1^m) = 1$, then $(\varphi, 1^m) \notin \mathrm{SAT}^*$. This means $\neg\varphi \notin \mathrm{SAT}$ and hence φ is a tautology. On the other hand every formula φ has a Q-proof if we choose the parameter m big enough.

Clearly Q is automatizable because $Q \vdash_{\leq m} \varphi$ implies in particular that $(\varphi, 1^m)$ is a Q-proof of φ.

P-proofs are translated to Q-proofs by the polynomial time computable function

$$\pi \mapsto (P(\pi), 1^{|\pi|}) \ ,$$

hence Q p-simulates P.

For the other direction let $Q \geq_p P$ be an automatizable proof system. Then $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ is p-separable. Because $P \leq_p Q$ we get

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(Q), \mathrm{SAT}^*) \ .$$

Hence also $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ is p-separable. □

This theorem indicates that instead of concentrating on automatizability it might be more important to investigate the p-separability of the canonical pairs. Therefore proof systems which have automatizable extensions $Q \geq_p P$ are called *weakly automatizable* (cf. (AB02)).

Although practical evidence seems to suggest that finding resolution proofs is easy, Alekhnovich and Razborov established in (AR01) the non-automatizability of resolution under an assumption from parameterized complexity ($\mathsf{W}[\mathsf{P}]$ is not tractable). The question whether resolution is weakly automatizable is still open. Atserias and Bonet (AB02) show that this question is equivalent to whether an extension of resolution $Res(2)$ has the efficient interpolation property (cf. Sect. 4.4.3).

Strong systems simulating Frege systems are known to be not automatizable under cryptographic assumptions. We will return to this problem in Sect. 5.1.

4.4.3 The Interpolation Pair and Feasible Interpolation

In this section we describe how the feasible interpolation property of a proof system can be modeled by a disjoint NP-pair. Feasible interpolation has been successfully used to show lower bounds to the proof size of a number of proof

systems like resolution and cutting planes. It originates in the classical interpolation theorem of Craig of which we only need the propositional version.

Theorem 4.4.6 (Craig's Interpolation Theorem (Cra57))
Let $\varphi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{y})$ be propositional formulas with all variables displayed. Let $\bar{y}$ and $\bar{z}$ be distinct tuples of variables such that $\bar{x}$ are the common variables of φ and ψ. If

$$\varphi(\bar{x}, \bar{y}) \to \psi(\bar{x}, \bar{z})$$

is a tautology, then there exists a propositional formula $\theta(\bar{x})$ using only the common variables of φ and ψ such that

$$\varphi(\bar{x}, \bar{y}) \to \theta(\bar{x}) \quad \text{and} \quad \theta(\bar{x}) \to \psi(\bar{x}, \bar{z})$$

are tautologies.

Proof. Consider the Boolean function $\exists\bar{y}\varphi(\bar{x}, \bar{y})$. This function interpolates $\varphi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{y})$ because

$$\varphi(\bar{x}, \bar{y}) \to \exists\bar{y}\varphi(\bar{x}, \bar{y})$$

is always a tautology and since $\varphi(\bar{x}, \bar{y}) \to \psi(\bar{x}, \bar{z})$ is tautological this is also true for

$$(\exists\bar{y}\varphi(\bar{x}, \bar{y})) \to \psi(\bar{x}, \bar{z}) \ .$$

Every Boolean function can be described by a propositional formula in the same variables. Hence any formula expressing $\exists\bar{y}\varphi(\bar{x}, \bar{y})$ is an interpolant of $\varphi(\bar{x}, \bar{y}) \to \psi(\bar{x}, \bar{z})$. Alternatively we could have taken a formula for $\forall\bar{z}\psi(\bar{x}, \bar{z})$. □

Next we define the feasible interpolation property.

Definition 4.4.7 (Krajíček (Kra97)) *A proof system P has* feasible interpolation *if there exists a polynomial time procedure that takes as input an implication $\varphi(\bar{x}, \bar{y}) \to \psi(\bar{x}, \bar{z})$ and a P-proof π of $\varphi(\bar{x}, \bar{y}) \to \psi(\bar{x}, \bar{z})$ and outputs a Boolean circuit $C(\bar{x})$ such that for every propositional assignment $\bar{a}$ the following holds:*

1. *If $\varphi(\bar{a}, \bar{y})$ is satisfiable, then $C(\bar{a})$ outputs 1.*
2. *If $\neg\psi(\bar{a}, \bar{z})$ is satisfiable, then $C(\bar{a})$ outputs 0.*

Feasible interpolation has been shown for resolution (Kra97), the cutting planes system (BPR97; Kra97; Pud97) and some algebraic proof systems (PS98). Combined with lower bounds for the separation of the clique colouring pair by monotone Boolean circuits (Raz85; AB87) these results yield lower bounds for the proof lengths of the above proof systems. We refer to the survey (Pud98) for a detailed presentation of this approach.

To capture the feasible interpolation property by disjoint NP-pairs Pudlák defines in (Pud03) an *interpolation pair* (I_P^0, I_P^1) for a proof system P. To be consistent with our notation of pairs we denote this pair by $(I_1(P), I_2(P))$. It is defined as follows:

$$\begin{aligned} I_1(P) &= \{(\varphi,\psi,\pi) \mid P(\pi) = \varphi \vee \psi,\ \mathrm{Var}(\varphi) \cap \mathrm{Var}(\psi) = \emptyset \text{ and } \neg\varphi \in \mathrm{SAT}\} \\ I_2(P) &= \{(\varphi,\psi,\pi) \mid P(\pi) = \varphi \vee \psi,\ \mathrm{Var}(\varphi) \cap \mathrm{Var}(\psi) = \emptyset \text{ and } \neg\psi \in \mathrm{SAT}\}. \end{aligned}$$

Whether or not a proof system admits feasible interpolation can be read off from the interpolation pair, namely:

Theorem 4.4.8 (Pudlák (Pud03)) *Let P be a propositional proof system that is efficiently closed under substitutions by constants. Likewise suppose we can efficiently modify a P-proof of an implication $\varphi \to \psi$ to a P-proof of $\neg\varphi \vee \psi$ and vice versa.*

Then $(I_1(P), I_2(P))$ is p-separable if and only if P has the feasible interpolation property.

Proof. Suppose $(I_1(P), I_2(P))$ is separated by the polynomial time computable function f, i.e.

$$\begin{aligned} (\varphi,\psi,\pi) \in I_1(P) \quad &\Longrightarrow \quad f(\varphi,\psi,\pi) = 1 \\ (\varphi,\psi,\pi) \in I_2(P) \quad &\Longrightarrow \quad f(\varphi,\psi,\pi) = 0 \ . \end{aligned}$$

Let the implication $\varphi(\bar{x},\bar{y}) \to \psi(\bar{x},\bar{z})$ be given together with a P-proof π of this formula. We have to construct a circuit C with inputs $\bar{x}$ that interpolates φ and ψ.

Because P can handle implications and is closed under substitutions by constants in an effective manner we get from the P-proof π of $\varphi(\bar{x},\bar{y}) \to \psi(\bar{x},\bar{z})$ an at most polynomially longer P-proof π' of $\neg\varphi(\bar{a},\bar{y}) \vee \psi(\bar{a},\bar{z})$ for constants $\bar{a}$. The circuit C on input $\bar{a}$ first computes this proof π' from the proof π which is hardwired into C and then evaluates the function

$$f(\neg\varphi(\bar{a},\bar{y}), \psi(\bar{a},\bar{z}), \pi') \ .$$

If $\varphi(\bar{a},\bar{y})$ is satisfiable, then f outputs 1, hence

$$\models \varphi(\bar{x},\bar{y}) \to C(\bar{x}) \ .$$

If $\neg\psi(\bar{a},\bar{z})$ is satisfiable, i.e. $\psi(\bar{a},\bar{z})$ is not tautological, then f outputs 0, hence

$$\models C(\bar{x}) \to \psi(\bar{x},\bar{z}) \ .$$

For the other direction suppose that P admits feasible interpolation. We need to construct a function f that separates $I_1(P)$ and $I_2(P)$. On input $(\varphi(\bar{y}),\psi(\bar{z}),\pi)$ we first check that φ and ψ have no common variables and that π is indeed a P-proof of $\varphi \vee \psi$. Then we construct from π a P-proof π' of $\neg\varphi(\bar{y}) \to \psi(\bar{z})$. Now feasible interpolation for P gives us a circuit without free inputs that interpolates $\neg\varphi(\bar{y})$ and $\psi(\bar{z})$. Evaluating this circuit we obtain the answer to the desired function f. □

It is not difficult to show that for proof systems with suitable closure properties the interpolation pair is $\leq_p$-reducible to the canonical pair of the proof system (cf. Proposition 4.8.1). Therefore automatizable proof systems also have the feasible interpolation property. The converse is probably not true, as resolution has feasible interpolation (Kra97) but is not believed to be automatizable (AR01).

4.5 Representations of NP-Pairs

In the previous section we explained how properties of propositional proof systems can be captured by disjoint NP-pairs that are suitably defined from these proof systems. Hence, exploring the theory of disjoint NP-pairs can help us to solve problems from propositional proof complexity. In Sect. 5.2 we will discuss another application of this kind.

Conversely, we now aim to transfer proof-theoretic knowledge to the theory of NP-pairs to gain a more detailed understanding of the structure of the class of disjoint NP-pairs and in particular of the NP-pairs defined from propositional proof systems. For this we need to represent arbitrary disjoint NP-pairs in propositional proof systems. This can be done uniformly in theories of bounded arithmetic or non-uniformly in propositional proof systems. We will start with the uniform concept which was first considered by Razborov (Raz94).

Definition 4.5.1 (Razborov (Raz94)) *A Σ^b_1-formula φ is an* arithmetic representation *of an* NP*-set A if for all natural numbers a*

$$\mathcal{N} \models \varphi(a) \iff a \in A \ .$$

A DNPP (A,B) is representable *in an L-theory T if there are Σ^b_1-formulas φ and ψ representing A and B, respectively, such that*

$$T \vdash (\forall x)(\neg\varphi(x) \vee \neg\psi(x)) \ .$$

By $\mathsf{DNPP}(T)$ *we denote the class of all disjoint* NP*-pairs that are representable in* T*.*

Since $(\forall x)(\neg\varphi(x) \vee \neg\psi(x))$ is a $\forall\Pi_1^b$-formula we can also express the disjointness of A and B propositionally by the sequence of tautologies $\|\neg\varphi(x) \vee \neg\psi(x)\|^n$. Hence propositional representations of disjoint NP-pairs can be simply obtained by transforming Definition 4.5.1 with the translation $\|.\|$ to the propositional level. However, we will give a more general definition. For this we first need to define a propositional encoding of NP-sets.

Definition 4.5.2 *Let* A *be an* NP*-set over the alphabet* $\{0,1\}$*. A* propositional representation *for* A *is a sequence of propositional formulas* $\varphi_n(\bar{x},\bar{y})$ *with the following properties:*

1. $\varphi_n(\bar{x},\bar{y})$ *has propositional variables* $\bar{x}$ *and* $\bar{y}$ *such that* $\bar{x}$ *is a vector of* n *propositional variables.*
2. *There exists a polynomial time algorithm that on input* 1^n *outputs* $\varphi_n(\bar{x},\bar{y})$*.*
3. *Let* $\bar{a} \in \{0,1\}^n$*. Then* $\bar{a} \in A$ *if and only if* $\varphi_n(\bar{a},\bar{y})$ *is satisfiable.*

Once we have a propositional description of NP-sets we can also represent disjoint NP-sets in propositional proof systems. This notion is captured by the next definition.

Definition 4.5.3 *Let* P *be a propositional proof system. A disjoint* NP*-pair* (A,B) *is* representable in P *if there are propositional representations* $\varphi_n(\bar{x},\bar{y})$ *of* A *and* $\psi_n(\bar{x},\bar{z})$ *of* B *such that* $\bar{x}$ *are the common variables of* $\varphi_n(\bar{x},\bar{y})$ *and* $\psi_n(\bar{x},\bar{z})$ *and*

$$P \vdash_* \neg\varphi_n(\bar{x},\bar{y}) \vee \neg\psi_n(\bar{x},\bar{z}) \ .$$

By $\mathsf{DNPP}(P)$ *we denote the class of all disjoint* NP*-pairs which are representable in* P*.*

In the class $\mathsf{DNPP}(P)$ we collect those NP-pairs for which the disjointness is efficiently provable in the proof system P. Clearly, considering stronger proof systems we expect this class to grow, namely:

Proposition 4.5.4 *Let* P *and* Q *be proof systems. If* $P \leq Q$*, then* $\mathsf{DNPP}(P) \subseteq \mathsf{DNPP}(Q)$*.*

This simple observation also implies that the representability of a disjoint NP-pair is a robust property, i.e. if P and Q are equivalent proof systems,

then a pair (A, B) is representable in P if and only if it is representable in Q.

We remark that the provability of the disjointness of a pair (A, B) in some proof system depends crucially on the choice of the representations for A and B.

Proposition 4.5.5 *Let P be proof system such that the system $EF + \mathrm{RFN}(P)$ is not optimal and let $(A, B) \in \mathsf{DNPP}(P)$. Then there exist representations φ_n of A and ψ_n of B such that $P \nvdash_* \neg\varphi_n \vee \neg\psi_n$.*

Proof. Let (A, B) be representable in P via the representations φ'_n and ψ'_n, i.e. $P \vdash_* \neg\varphi'_n \vee \neg\psi'_n$. By Q we denote the proof system $EF + \mathrm{RFN}(P)$. Because Q is not optimal and fulfills all conditions from Theorem 3.7.8 we can use this theorem to get a sequence τ_n of hard tautologies for Q. We define

$$\begin{aligned} \varphi_n(\bar{x}, \bar{y}, \bar{u}) &= \varphi'_n(\bar{x}, \bar{y}) \vee \neg\tau_n(\bar{u}) \\ \psi_n(\bar{x}, \bar{z}, \bar{v}) &= \psi'_n(\bar{x}, \bar{z}) \vee \neg\tau_n(\bar{v}) \end{aligned}$$

where all tuples of variables $\bar{x}$, $\bar{y}$, $\bar{z}$, $\bar{u}$ and $\bar{v}$ are pairwise disjoint. As $\neg\tau_n(\bar{u})$ is not satisfiable $\varphi'_n(\bar{x}, \bar{y}) \vee \neg\tau_n(\bar{u})$ represents A. Similarly, ψ_n is a propositional representation for B. But Q does not prove the disjointness of A and B with respect to the representations φ_n and ψ_n. Assume on the contrary that

$$Q \vdash_* \neg\varphi_n \vee \neg\psi_n \ .$$

By definition this means

$$Q \vdash_* \neg(\varphi'_n(\bar{x}, \bar{y}) \vee \neg\tau_n(\bar{u})) \ \vee \ \neg(\psi'_n(\bar{x}, \bar{z}) \vee \neg\tau_n(\bar{v})) \ .$$

As Q can perform basic operations with formulas we get polynomial size Q-proofs of

$$\begin{aligned} (\neg\varphi'_n(\bar{x}, \bar{y}) \vee \neg\psi'_n(\bar{x}, \bar{z})) \ \wedge \ (\neg\varphi'_n(\bar{x}, \bar{y}) \vee \tau_n(\bar{v})) \wedge \\ (\neg\psi'_n(\bar{x}, \bar{z}) \vee \tau_n(\bar{u})) \ \wedge \ (\tau_n(\bar{u}) \vee \tau_n(\bar{v})) \ . \end{aligned}$$

Because Q is closed under conjunctions we obtain

$$Q \vdash_* \tau_n(\bar{u}) \vee \tau_n(\bar{v}) \ .$$

As these are two identical copies of the same formula with disjoint variables we can prove in EF the formula $\tau_n(\bar{u})$ by substituting the variables $\bar{v}$ by $\bar{u}$.

Hence we derive $Q \vdash_* \tau_n(\bar{u})$, contradicting the choice of τ_n as hard tautologies for Q. Thus we have shown

$$Q \nvdash_* \neg\varphi_n \vee \neg\psi_n$$

and because $P \leq Q$ we have also proven our claim $P \nvdash_* \neg\varphi_n \vee \neg\psi_n$. □

Clearly, if optimal systems do not exist, then we have hard tautologies for all the systems $EF + \mathrm{RFN}(P)$. Hence we get:

Corollary 4.5.6 *If optimal proof systems do not exist, then the following holds: for every proof system P and for every disjoint* NP*-pair (A, B) there exist propositional representations φ_n for A and ψ_n for B such that P does not prove the disjointness of (A, B) with respect to these representations, i.e. $P \nvdash_* \neg\varphi_n \vee \neg\psi_n$.*

Let us give a concrete example for this situation. In (Pud99) Pudlák shows that the disjointness of the Clique-Colouring pair is not provable with polynomial size proofs in the cutting planes system CP for some canonical representations of the components CC_0 and CC_1. As CP simulates resolution the disjointness of (CC_0, CC_1) is also not provable in resolution with respect to these representations. On the other hand, the Clique-Colouring pair is p-separable as shown by Lovász (Lov79). Hence (CC_0, CC_1) is contained in $\mathsf{DNPP}(Res)$ as the following argument shows. We choose some simple p-separable pair (A, B) that is representable in resolution. As all p-separable pairs are equivalent we can reduce (CC_0, CC_1) to (A, B). The class $\mathsf{DNPP}(Res)$ is closed under $\leq_p$-reductions (we will show this in Sect. 4.6, Corollary 4.6.2). Therefore we get $(CC_0, CC_1) \in \mathsf{DNPP}(Res)$ which means that there exist polynomial size resolution proofs for the disjointness of the Clique-Colouring pair for suitable representations of its components.

Now we will compare the uniform and non-uniform representations. We first show that the NP-pairs representable in a strong proof system are also representable in the corresponding theory.

Proposition 4.5.7 *Let $T \supseteq S^1_2$ be an L-theory and let P be a proof system that is closed under substitutions by constants. Then $T \vdash \mathrm{RFN}(P)$ implies $\mathsf{DNPP}(P) \subseteq \mathsf{DNPP}(T)$.*

Proof. Let (A, B) be a disjoint NP-pair in $\mathsf{DNPP}(P)$ and let $\varphi_n(\bar{x}, \bar{y})$ and $\psi_n(\bar{x}, \bar{z})$ be propositional representations for A and B, respectively, such that

$$P \vdash_* \neg\varphi_n(\bar{x}, \bar{y}) \vee \neg\psi_n(\bar{x}, \bar{z}) \ .$$

Because P is closed under substitutions by constants there exists a polynomial p such that for all $\bar{a} \in \{0,1\}^n$

$$P \vdash_{\leq p(n)} \neg\varphi_n(\bar{a},\bar{y}) \vee \neg\psi_n(\bar{a},\bar{z}) \ .$$

Assume further that the polynomial time computable functions f and g generate the formulas φ_n and ψ_n, i.e.

$$f(1^n) = \varphi_n(\bar{x},\bar{y}) \quad \text{and} \quad g(1^n) = \psi_n(\bar{x},\bar{z}) \ .$$

Consider the first-order formula

$$\varphi(\alpha) = Assign(\alpha,\bar{x}) \wedge \neg\mathrm{Taut}(\neg f(1^{|\alpha|})(\alpha(\bar{x}),\bar{y})) \ .$$

As this notation is not completely precise let us explain how to understand the definition of φ. At input $1^{|\alpha|}$ the function f outputs the formula $\varphi_{|\alpha|}(\bar{x},\bar{y})$. In φ the computation of f is expressed by a Σ^b_1-formula. Then we use again the free variable α of φ to obtain a propositional assignment to the propositional variables $\bar{x}$. The formula $\neg\mathrm{Taut}(\neg f(1^{|\alpha|})(\alpha(\bar{x}),\bar{y}))$ is a Σ^b_1-formulation for the unsatisfiability of $\varphi_{|\alpha|}(\bar{x},\bar{y})$, where the variables $\bar{x}$ are substituted by the constants specified in α and only the variables $\bar{y}$ remain free.

The above explanation shows that φ is a Σ^b_1-formula. Moreover, it is clear that φ represents A. Similarly, we define a representation for B as

$$\begin{aligned}\psi(\alpha) \ = \ & Assign(\alpha,\bar{x}) \wedge \neg\mathrm{Taut}(\neg g(1^{|\alpha|})(\alpha(\bar{x}),\bar{z})) \wedge \\ & (\exists\pi)|\pi| \leq p(|\alpha|) \wedge \mathrm{Prf}_P(\pi, \neg f(1^{|\alpha|})(\alpha(\bar{x}),\bar{y}) \vee \neg g(1^{|\alpha|})(\alpha(\bar{x}),\bar{z})) \ .\end{aligned}$$

Let us first verify that $\psi \in \Sigma^b_1$. The first line of the definition of φ is Σ^b_1 analogously as in the definition of φ. As Prf_P has a Δ^b_1-definition in S^1_2 and $T \supseteq S^1_2$ also the second line can be given a Σ^b_1-formulation, and hence $\psi \in \Sigma^b_1$.

Let $\bar{a} \in \{0,1\}^{|\alpha|}$ be the tupel of constants specified by the assignment α. Then the first line of the definition of ψ expresses $\bar{a} \in B$ analogously as in the definition of φ. Because

$$\neg f(1^{|\alpha|})(\alpha(\bar{x}),\bar{y}) \vee \neg g(1^{|\alpha|})(\alpha(\bar{x}),\bar{z})$$

equals the formula

$$\neg\varphi_{|\alpha|}(\bar{a},\bar{y}) \vee \neg\psi_{|\alpha|}(\bar{a},\bar{z})$$

which by assumption has a P-proof of length $\leq p(|\alpha|)$ also the second part of ψ is fulfilled for $\bar{a} \in B$. Therefore ψ represents B.

It remains to verify that T can prove the disjointness of A and B with respect to the above representations. For this assume that M is a model of

T and $\alpha \in M$ is an element such that $M \models \psi(\alpha)$. In particular this means that there exists an element $\pi \in M$ such that

$$M \models \mathrm{Prf}_P(\pi, \neg f(1^{|\alpha|})(\alpha(\bar{x}), \bar{y}) \vee \neg g(1^{|\alpha|})(\alpha(\bar{x}), \bar{z})) \ .$$

Because $T \vdash \mathrm{RFN}(P)$ this implies

$$M \models \mathrm{Taut}(\neg f(1^{|\alpha|})(\alpha(\bar{x}), \bar{y}) \vee \neg g(1^{|\alpha|})(\alpha(\bar{x}), \bar{z})) \ .$$

The theory $T \supseteq S^1_2$ is strong enough to prove Tarski's truth conditions for the propositional satisfaction relation $\models$ (cf. (Kra95) Lemma 9.3.9). In particular T proves

$$(\forall \varphi, \psi, \alpha) Assign(\alpha, \varphi \vee \psi) \wedge (\alpha \models \varphi \vee \psi) \rightarrow (\alpha \models \varphi) \vee (\alpha \models \psi) \ .$$

Therefore T proves that a tautological disjunction of formulas without common variables contains at least one tautological disjunct, and hence we get

$$M \models \mathrm{Taut}(\neg f(1^{|\alpha|})(\alpha(\bar{x}), \bar{y})) \vee \mathrm{Taut}(\neg g(1^{|\alpha|})(\alpha(\bar{x}), \bar{z})) \ .$$

But because $M \models \psi(\alpha)$ we also have

$$M \models \neg \mathrm{Taut}(\neg g(1^{|\alpha|})(\alpha(\bar{x}), \bar{z}))$$

implying

$$M \models \mathrm{Taut}(\neg f(1^{|\alpha|})(\alpha(\bar{x}), \bar{y})) \ ,$$

and therefore $M \not\models \varphi(\alpha)$. Hence we have shown $T \vdash (\forall x) \neg \varphi(x) \vee \neg \psi(x)$. □

Next we prove that for regular proof systems also the other inclusion is valid, yielding equality between the classes $\mathsf{DNPP}(P)$ and $\mathsf{DNPP}(T)$.

Theorem 4.5.8 *Let $P \geq EF$ be a regular proof system which is closed under substitutions by constants and let $T \supseteq S^1_2$ be a theory corresponding to T. Then $\mathsf{DNPP}(P) = \mathsf{DNPP}(T)$.*

Proof. The first inclusion was already proven in Proposition 4.5.7. To show $\mathsf{DNPP}(T) \subseteq \mathsf{DNPP}(P)$ let φ and ψ be Σ^b_1-formulas representing A and B, respectively, such that

$$T \vdash (\forall x) \neg \varphi(x) \vee \neg \psi(x) \ . \tag{4.1}$$

We define the propositional representations of A and B as the $\|.\|$-translations of φ and ψ, namely

$$\varphi_n(\bar{x}, \bar{y}) = \|\varphi(x)\|^n \quad \text{and} \quad \psi_n(\bar{x}, \bar{z}) = \|\psi(x)\|^n$$

where we choose the auxiliary variables $\bar{y}$ of $\|\varphi(x)\|^n$ and $\bar{z}$ of $\|\psi(x)\|^n$ disjoint. These sequences can be generated in polynomial time and hence represent A and B by Theorem 3.2.1. Because the formula $(\forall x)\neg\varphi(x) \vee \neg\psi(x)$ in equation (4.1) is a $\forall\Pi^b_1$-formula we derive

$$P \vdash_* \|\neg\varphi(x) \vee \neg\psi(x)\|^n \ ,$$

which implies

$$P \vdash_* \neg\|\varphi(x)\|^n \vee \neg\|\psi(x)\|^n \ .$$

□

At first sight Theorem 4.5.8 might come as a surprise as it states that the non-uniform and uniform concepts equal when representing disjoint NP-pairs in regular proof systems. The uniform representations of NP-pairs are translated via $\|.\|$ to non-uniform representations in a straightforward manner. For the transformation of propositional representations into first-order formulas as in Proposition 4.5.7 it is, however, necessary to essentially change the representation of one of the components (in the proof of Proposition 4.5.7 of that of B).

4.6 The Complexity Class DNPP(P)

The aim of this section is to show that the subclasses DNPP(P) of DisjNP as defined in the last section are indeed examples for well defined complexity classes. We will provide justification for this claim by demonstrating that the classes DNPP(P) are closed under reductions and also posses hard or complete pairs for well defined proof systems P.

We start by giving sufficient conditions for the closure of DNPP(P) under $\leq_p$ (and hence also under $\leq_s$). Translating the reductions to the propositional level we have to work with uniform circuit families computing the reduction functions. Since it is possible in resolution to prove the uniqueness of circuit computations we can show the following:

Proposition 4.6.1 *Let P be a proof system which simulates resolution and is closed under disjunctions. Then* DNPP(P) *is closed under $\leq_p$.*

Proof. Let (A, B) and (C, D) be disjoint NP-pairs. Let (C, D) be representable in P, i.e. there exist representations $\varphi_n(\bar{x}, \bar{y})$ and $\psi_n(\bar{x}, \bar{z})$ of C and D, respectively, such that

$$P \vdash_* \neg\varphi_n(\bar{x}, \bar{y}) \vee \neg\psi_n(\bar{x}, \bar{z}) \ .$$

Assume further that (A, B) is $\leq_p$-reducible to (C, D) via the polynomial time computable function f. We have to show that also (A, B) is representable in P. For this we fix arbitrary representations $\chi_n(\bar{x}, \bar{r})$ and $\theta_n(\bar{x}, \bar{s})$ for A and B, respectively. Without loss of generality we may assume that the reduction function f generates on inputs of length n outputs of length exactly $p(n)$ for some fixed polynomial p. This can be achieved for example by adding leading zeros to outputs of length $\leq p(n)$. Let

$$C_n : \{0,1\}^n \to \{0,1\}^{p(n)}$$

be a uniform circuit family which computes the function f. The computation of the circuits C_n can be described by propositional formulas $C_n(\bar{x}, \bar{p}, \bar{u})$ which state that on input corresponding to the propositional variables $\bar{x}$ the circuit produces the output corresponding to $\bar{p}$. The variables $\bar{u}$ are auxiliary variables for the gates of the circuit.

Consider the sequence of propositional formulas

$$\chi_n(\bar{x}, \bar{r}) \wedge C_n(\bar{x}, \bar{p}, \bar{u}) \wedge \varphi_{p(n)}(\bar{p}, \bar{y}) \ . \tag{4.2}$$

These formulas provide a propositional representation of the set A because they propositionally express that $\bar{x} \in A$ and there exists a computation of C_n on input $\bar{x}$ that outputs an element from the set C. Similarly, the sequence

$$\theta_n(\bar{x}, \bar{s}) \wedge C_n(\bar{x}, \bar{q}, \bar{v}) \wedge \psi_{p(n)}(\bar{q}, \bar{z}) \tag{4.3}$$

represents B. We have to check that P proves the disjointness of A and B with respect to these representations. The P-proof proceeds along the following lines. By hypothesis we have polynomial size P-proofs for the formulas

$$\neg\varphi_{p(n)}(\bar{p}, \bar{y}) \vee \neg\psi_{p(n)}(\bar{p}, \bar{z}) \ . \tag{4.4}$$

By induction on the number of gates of a circuit we can show that resolution proves the uniqueness of computations of Boolean circuits in polynomial size resolution proofs. Because P simulates resolution this means that we have polynomial size P-proofs of the formulas

$$C_n(\bar{x}, \bar{p}, \bar{u}) \wedge C_n(\bar{x}, \bar{q}, \bar{v}) \ \to \ (\bar{p} \leftrightarrow \bar{q}) \ . \tag{4.5}$$

From (4.4) and (4.5) we obtain polynomial size P-proofs of

$$C_n(\bar{x}, \bar{p}, \bar{u}) \wedge C_n(\bar{x}, \bar{q}, \bar{v}) \ \to \ \neg\varphi_{p(n)}(\bar{p}, \bar{y}) \vee \neg\psi_{p(n)}(\bar{q}, \bar{z}) \ . \tag{4.6}$$

Because P is closed under disjunctions we get from (4.6) polynomial size P-proofs of

$$\neg\chi_n(\bar{x}, \bar{r}) \vee \neg\theta_n(\bar{x}, \bar{s}) \vee \neg C_n(\bar{x}, \bar{p}, \bar{u}) \vee \neg C_n(\bar{x}, \bar{q}, \bar{v}) \vee \neg\varphi_{p(n)}(\bar{p}, \bar{y}) \vee \neg\psi_{p(n)}(\bar{q}, \bar{z}) \ .$$

But this exactly means that P proves the disjointness of A and B with respect to the propositional representations (4.2) and (4.3). Hence $(A, B) \in \mathsf{DNPP}(P)$. □

We instantiate Proposition 4.6.1 for our standard examples of proof systems:

Corollary 4.6.2 *The class* $\mathsf{DNPP}(P)$ *is closed under* $\leq_p$ *and* $\leq_s$ *for the following proof systems* P*: resolution, Frege systems and all systems* $EF + \Phi$ *for polynomial time computable sets* $\Phi \subseteq \mathrm{TAUT}$*.*

Next we show the hardness of the canonical pair of a proof system P for the class $\mathsf{DNPP}(P)$.

Theorem 4.6.3 *Let* P *be a proof system that is closed under substitutions by constants and modus ponens and can evaluate formulas without variables. Then* $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ *is* $\leq_p$*-hard for* $\mathsf{DNPP}(P)$*.*

Proof. Let (A, B) be a DNPP and let $\varphi_n(\bar{x}, \bar{y})$ and $\psi_n(\bar{x}, \bar{z})$ be propositional representations of A and B, respectively, such that

$$P \vdash_* \neg\varphi_n(\bar{x}, \bar{y}) \vee \neg\psi_n(\bar{x}, \bar{z}) \ .$$

We have to show that

$$(A, B) \ \leq_p \ (\mathrm{Ref}(P), \mathrm{SAT}^*) \ .$$

We claim that the reduction is given by

$$a \ \mapsto \ (\neg\psi_{|a|}(\bar{a}, \bar{z}), 1^{p(|a|)})$$

for some suitable polynomial p. To see the correctness of the reduction let first be $a \in A$. Then there exists a witness $\bar{b}$ such that $\models \varphi_{|a|}(\bar{a}, \bar{b})$. From the P-proof of $\neg\varphi_{|a|}(\bar{x}, \bar{y}) \vee \neg\psi_{|a|}(\bar{x}, \bar{z})$ we get by substituting $\bar{a}$ for $\bar{x}$ and $\bar{b}$ for $\bar{y}$ a polynomially longer P-proof of $\neg\varphi_{|a|}(\bar{a}, \bar{b}) \vee \neg\psi_{|a|}(\bar{a}, \bar{z})$. $\neg\varphi_{|a|}(\bar{a}, \bar{b})$ is a false propositional formula without free variables and hence can be refuted with polynomial size P-proofs. An application of modus ponens gives a P-proof of $\neg\psi_{|a|}(\bar{a}, \bar{z})$ as desired.

Assume now $a \in B$. Then $\neg\neg\psi_{|a|}(\bar{a}, \bar{z}) = \psi_{|a|}(\bar{a}, \bar{z})$ is satisfiable and hence $(\neg\psi_{|a|}(\bar{a}, \bar{z}), 1^{p(|a|)}) \in \mathrm{SAT}^*$. □

4.7 The Canonical Pair and the Reflection Principle

In this section we turn to proof systems that have the reflection property. We first show that the reflection property of a proof system corresponds to the representability of the canonical pair in the proof system. This is another example for the characterization of proof-theoretic properties by disjoint NP-pairs. The link between the canonical pair and the reflection property is already apparent from the definition of $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ and is also discussed in (Pud03). Using our terminology from Sect. 4.5 we may phrase this connection precisely as:

Proposition 4.7.1 *Let P be a proof system. Then P has the reflection property if and only if the canonical pair of P is representable in P with respect to the standard representations of* $\mathrm{Ref}(P)$ *and* SAT^*.

Proof. By the standard representations of $\mathrm{Ref}(P)$ and SAT^* we mean the $\|.\|$-translations of the first-order formulas

$$(\exists \pi)\ |\pi| \leq m \wedge \mathrm{Prf}_P(\pi, \varphi)$$

for $\mathrm{Ref}(P)$ and

$$(\exists \alpha)\ |\alpha| \leq |\varphi| \wedge \alpha \models \neg\varphi$$

for SAT^*. The representability of $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ with respect to these representations means

$$P \vdash_* \|(\varphi, 1^m) \notin \mathrm{Ref}(P) \vee (\varphi, 1^m) \notin \mathrm{SAT}^*)\|^{n,m} \ ,$$

i.e.

$$P \vdash_* \|\neg \mathrm{Prf}_P(\pi, \varphi) \vee \alpha \not\models \neg\varphi\|^{n,m} \ .$$

$(\forall \alpha)\ |\alpha| \leq |\varphi| \wedge \alpha \not\models \neg\varphi$ is equivalent to $\mathrm{Taut}(\varphi)$, hence

$$P \vdash_* \|\neg \mathrm{Prf}_P(\pi, \varphi) \vee \mathrm{Taut}(\varphi)\|^{n,m} \ ,$$

i.e.

$$P \vdash_* \|\mathrm{Prf}_P(\pi, \varphi) \rightarrow \mathrm{Taut}(\varphi)\|^{n,m} \ ,$$

which is by definition $P \vdash_* \|\mathrm{RFN}(P)\|$. □

Using Proposition 4.7.1 we conclude from Theorem 4.6.3 the following:

Corollary 4.7.2 *Let P be a proof system that has the reflection property. Assume further that P is closed under substitutions by constants and modus ponens and can evaluate formulas without variables. Then* $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ *is $\leq_p$-complete for* $\mathsf{DNPP}(P)$.

By Theorem 3.6.9 this means for our standard examples of proof systems:

Corollary 4.7.3 *Let Φ be a polynomial time decidable set of true Π_1^b-formulas. Then $(\mathrm{Ref}(EF+\|\Phi\|), \mathrm{SAT}^*)$ is $\leq_p$-complete for $\mathsf{DNPP}(EF+\|\Phi\|)$.*

What is actually needed for Corollary 4.7.2 is not the reflection property of P but the representability of $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ in the proof system P. We already remarked that reflection for P implies $(\mathrm{Ref}(P), \mathrm{SAT}^*) \in \mathsf{DNPP}(P)$. However, the next proposition shows that the provability of the reflection principle of a system and the representability of its canonical pair are different concepts.

Proposition 4.7.4 *Let P be a proof system of the form $EF + \Phi$ for polynomial time computable $\Phi \subseteq \mathrm{TAUT}$. Let further Q be a proof system such that*

$$Q \not\leq P \quad \textit{but} \quad (\mathrm{Ref}(Q), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(P), \mathrm{SAT}^*) \ .$$

Then $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ is representable in P but $P \not\vdash_* \|\mathrm{RFN}(Q)\|^n$.

Proof. Suppose the polynomial time computable function f performs the $\leq_p$-reduction from $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ to $(\mathrm{Ref}(P), \mathrm{SAT}^*)$. From this we conclude with Propositions 4.6.1 and 4.7.1 the representability of $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ in P. Going back to the proof of Proposition 4.6.1 we see that P proves the disjointness of $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ with respect to the following representations:

$$\mathrm{Ref}(Q) = \{(\varphi, 1^m) \mid (\varphi, 1^m) \in \mathrm{Ref}(Q) \text{ and } f(\varphi, 1^m) \in \mathrm{Ref}(P)\}$$

and

$$\mathrm{SAT}^* = \{(\varphi, 1^m) \mid (\varphi, 1^m) \in \mathrm{SAT}^* \text{ and } f(\varphi, 1^m) \in \mathrm{SAT}^*\} \ .$$

But if P proves the disjointness of $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ with respect to the standard representations

$$\mathrm{Ref}(Q) = \{(\varphi, 1^m) \mid (\exists \pi)\ |\pi| \leq m \wedge \mathrm{Prf}_P(\pi, \varphi)\}$$

and

$$\mathrm{SAT}^* = \{(\varphi, 1^m) \mid (\exists \alpha)\ |\alpha| \leq |\varphi| \wedge \alpha \models \neg\varphi\}$$

this means $P \vdash_* \|\mathrm{RFN}(Q)\|$ and by Lemma 3.7.6 we get $Q \leq P$ in contradiction to the hypothesis $Q \not\leq P$. □

In Sect. 4.13 we will show how to construct non-equivalent proof systems P, Q with equivalent canonical pairs which are needed for the hypothesis of Proposition 4.7.4.

In this context it is natural to ask whether the canonical pair of the resolution calculus Res is $\leq_p$-complete for $\mathsf{DNPP}(Res)$. In view of Corollary 4.7.2 and the above discussion knowing whether $(\mathrm{Ref}(Res), \mathrm{SAT}^*)$ is representable in resolution would answer this question. Atserias and Bonet (AB02) proved that resolution does not have the reflection property. By Proposition 4.7.1 this means that the disjointness of $(\mathrm{Ref}(Res), \mathrm{SAT}^*)$ is not provable in resolution with respect to the standard representation. However, we cannot exclude the possibility that we have short resolution proofs of the disjointness of $(\mathrm{Ref}(Res), \mathrm{SAT}^*)$ with respect to some other representation. At least we can remark that, unless the canonical pair of resolution is p-separable, these proofs would have to be essentially non-uniform.

Proposition 4.7.5 *If the canonical pair of resolution is not p-separable, then there do not exist proofs for the disjointness of* $(\mathrm{Ref}(Res), \mathrm{SAT}^*)$ *that can be generated in polynomial time.*

Proof. Assume on the contrary that $\varphi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ are representations of $\mathrm{Ref}(Res)$ and SAT^*, respectively, such that we can generate resolution proofs of $\neg\varphi(\bar{x}, \bar{y}) \vee \neg\psi(\bar{x}, \bar{z})$ in polynomial time. Because resolution has the feasible interpolation property (Kra97) this gives a polynomial time computable algorithm that on input 1^n produces a circuit $C_n(\bar{x})$ such that for all $\bar{a} \in \{0,1\}^n$

$$\begin{array}{lcl} \varphi(\bar{a}, \bar{y}) \text{ is satisfiable} & \Longrightarrow & C_n(\bar{a}) = 1 \\ \psi(\bar{a}, \bar{z}) \text{ is satisfiable} & \Longrightarrow & C_n(\bar{a}) = 0 \ . \end{array}$$

As φ and ψ are representations for $\mathrm{Ref}(Res)$ and SAT^*, respectively, this means that by evaluating the circuit C_n we get a separator for $(\mathrm{Ref}(Res), \mathrm{SAT}^*)$. Hence the canonical pair of resolution is p-separable. □

4.8 The Class $\mathsf{DNPP}(P)$ Under the Strong $\leq_s$-Reduction

In this section we will analyse the class $\mathsf{DNPP}(P)$ under the strong reduction $\leq_s$. This is interesting because we know that $\leq_s$ is indeed a proper refinement of $\leq_p$ (cf. Sect. 4.1). We start by associating to every proof system P a disjoint NP-pair $(U_1(P), U_2)$:

$$\begin{array}{rcl} U_1(P) & = & \{(\varphi, \psi, 1^m) \mid \mathrm{Var}(\varphi) \cap \mathrm{Var}(\psi) = \emptyset,\ \neg\varphi \in \mathrm{SAT} \text{ and } P \vdash_{\leq m} \varphi \vee \psi\} \\ U_2 & = & \{(\varphi, \psi, 1^m) \mid \mathrm{Var}(\varphi) \cap \mathrm{Var}(\psi) = \emptyset \text{ and } \neg\psi \in \mathrm{SAT}\} \ . \end{array}$$

In the following we will simply refer to this pair as the U-pair. Let us first argue that $(U_1(P), U_2)$ is indeed a disjoint NP-pair. Clearly both components are in NP. Let $(\varphi, \psi, 1^m) \in U_1(P)$. Since we have a P-proof of $\varphi \vee \psi$ the formula is a tautology. Because φ and ψ do not share variables one of φ or ψ is itself a tautology. Because $\neg\varphi$ is satisfiable ψ is a tautology. Therefore $\neg\psi \notin \mathrm{SAT}$ and hence $(\varphi, \psi, 1^m) \notin U_2$.

We could have defined the pair in a more symmetric way by requiring $P \vdash_{\leq m} \varphi \vee \psi$ also for the second component but for the following this is not important.

The U-pair is reminiscent of the interpolation pair $(I_1(P), I_2(P))$, the essential difference being that $(I_1(P), I_2(P))$ contains actual P-proofs while $(U_1(P), U_2)$ contains only information on their lengths. In the following we will show that both these pairs have similar function for $\mathsf{DNPP}(P)$ under $\leq_s$ as the canonical pairs have under the weaker reduction $\leq_p$. But before we come to this we need to compare $(U_1(P), U_2)$ with the canonical pair of P.

Proposition 4.8.1 *1. Let P be a proof system that is closed under disjunctions. Then $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (U_1(P), U_2)$.*

2. Let P be a proof system that is closed under substitutions by constants and modus ponens and evaluates formulas without variables. Then $(U_1(P), U_2) \leq_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$ and $(I_1(P), I_2(P)) \leq_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$.

Proof. The first reduction is given by

$$(\varphi, 1^m) \mapsto (\bot, \varphi, 1^{p(m)})$$

for a suitable polynomial p. To verify the correctness of the reduction let first $(\varphi, 1^m) \in \mathrm{Ref}(P)$. This means that $P \vdash_{\leq m} \varphi$ and because P is closed under disjunctions we infer $P \vdash_{\leq p(m)} \varphi \vee \bot$ for the respective polynomial p. We assume that the variables of φ and $\bot$ are chosen disjoint and since $\neg\bot = \top$ is satisfiable we get $(\bot, \varphi, 1^{p(m)}) \in U_1(P)$.

If $(\varphi, 1^m) \in \mathrm{SAT}^*$, then $\neg\varphi$ is satisfiable, hence $(\bot, \varphi, 1^{p(m)}) \in U_2$.

The reduction in part 2 of this proposition is performed by

$$(\varphi, \psi, 1^m) \mapsto (\psi, 1^{p(m)})$$

for some suitable polynomial p depending on the proof system P.

To verify the reduction let first $(\varphi(\bar{x}), \psi(\bar{y}), 1^m) \in U_1(P)$. Then $P \vdash_{\leq m} \varphi(\bar{x}) \vee \psi(\bar{y})$ and $\neg\varphi(\bar{x}) \in \mathrm{SAT}$. Choose a satisfying assignment $\bar{a}$ for $\neg\varphi(\bar{x})$. Because P is closed under substitutions by constants we get polynomially long P-proofs of $\varphi(\bar{a}) \vee \psi(\bar{y})$. $\varphi(\bar{a})$ is a false propositional formula without

variables which can be evaluated in P to $\bot$ in polynomially long proofs. Using modus ponens we obtain a P-proof of $\psi(\bar{y})$.

If $(\varphi, \psi, 1^m) \in U_2$, then $\neg\psi \in \mathrm{SAT}$ and hence $(\psi, 1^m) \in \mathrm{SAT}^*$.

The reduction from the interpolation pair to the canonical pair follows from combining the reduction from $(I_1(P), I_2(P))$ to $(U_1(P), U_2)$, given by $(\varphi, \psi, \pi) \mapsto (\varphi, \psi, 1^{|\pi|})$, with the previous reduction from $(U_1(P), U_2)$ to the canonical pair of P. □

As the proof systems $EF+\Phi$ for polynomial time computable $\Phi \subseteq \mathrm{TAUT}$ have all the properties listed in Proposition 4.8.1 we obtain:

Corollary 4.8.2 *Let $\Phi \subseteq \mathrm{TAUT}$ be computable in polynomial time. Then*

$$(\mathrm{Ref}(EF+\Phi), \mathrm{SAT}^*) \equiv_p (U_1(EF+\Phi), U_2) \ .$$

The following theorem is an analogon of Theorem 4.6.3 for the strong reduction $\leq_s$.

Theorem 4.8.3 *Let P be a proof system that is closed under substitutions by constants. Then $(U_1(P), U_2)$ is $\leq_s$-hard for $\mathsf{DNPP}(P)$.*

Proof. Let (A, B) be a DNPP and let $\varphi_n(\bar{x}, \bar{y})$ and $\psi_n(\bar{x}, \bar{z})$ be propositional representations of A and B, respectively, such that

$$P \ \vdash_* \ \neg\varphi_n(\bar{x}, \bar{y}) \vee \neg\psi_n(\bar{x}, \bar{z}) \ .$$

We claim that there exists a polynomial p such that

$$a \ \mapsto \ (\neg\varphi_{|a|}(\bar{a}, \bar{y}), \neg\psi_{|a|}(\bar{a}, \bar{z}), 1^{p(|a|)})$$

realizes a $\leq_s$-reduction from (A, B) to $(U_1(P), U_2)$.

Let first a be an element from A of length n. Because $\varphi_n(\bar{x}, \bar{y})$ represents A the formula $\varphi_n(\bar{a}, \bar{y})$ is satisfiable. As P is closed under substitutions by constants we have

$$P \ \vdash_{\leq p(n)} \ \neg\varphi_n(\bar{a}, \bar{y}) \vee \neg\psi_n(\bar{a}, \bar{z})$$

for the appropriate polynomial p. This confirms that

$$(\neg\varphi_n(\bar{a}, \bar{y}), \neg\psi_n(\bar{a}, \bar{z}), 1^{p(n)}) \in U_1(P) \ .$$

If $a \in B$, then $\psi_{|a|}(\bar{a}, \bar{z})$ is satisfiable and hence

$$(\neg\varphi_{|a|}(\bar{a}, \bar{y}), \neg\psi_{|a|}(\bar{a}, \bar{z}), 1^{p(|a|)}) \in U_2 \ .$$

If $a \notin A \cup B$, then neither $\varphi_{|a|}(\bar{a}, \bar{y})$ nor $\psi_{|a|}(\bar{a}, \bar{z})$ is satisfiable and hence $(\neg\varphi_{|a|}(\bar{a}, \bar{z}), \neg\psi_{|a|}(\bar{a}, \bar{z}), 1^{p(|a|)}) \notin U_1(P) \cup U_2$. □

As in the case of $\leq_p$ we can improve this hardness result to a completeness result for proof systems which have the reflection property.

Corollary 4.8.4 *Let P be a proof system that has the reflection property. Assume further that P is closed under substitutions by constants, modus ponens and disjunctions and can evaluate formulas without variables. Then $(U_1(P), U_2)$ is $\leq_s$-complete for $\mathsf{DNPP}(P)$.*

Proof. We use the reduction from $(U_1(P), U_2)$ to $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ as given by Proposition 4.8.1 to infer with Propositions 4.6.1 and 4.7.1 that $(U_1(P), U_2)$ is representable in P. Together with Theorem 4.8.3 this yields the $\leq_s$-completeness of $(U_1(P), U_2)$ for $\mathsf{DNPP}(P)$. □

For proof systems P corresponding to theories of bounded arithmetic we can also prove the $\leq_s$-completeness of the interpolation pair of P for $\mathsf{DNPP}(P)$. We first need to show that for regular proof systems P the pairs $(U_1(P), U_2)$ and $(I_1(P), I_2(P))$ are contained in $\mathsf{DNPP}(P)$.

Lemma 4.8.5 *Let $P \geq EF$ be a regular proof system. Then $(U_1(P), U_2)$ and $(I_1(P), I_2(P))$ are representable in P.*

Proof. Let $T \supseteq S^1_2$ be the theory corresponding to P. We first show that $(U_1(P), U_2)$ is representable in T via some standard representations using the formulas Prf_P and Taut. From this the representability of $(U_1(P), U_2)$ in P follows by Theorem 4.5.8.

Consider the first-order formulas

$$\begin{aligned}\theta(x,y,z) &= Form(x) \wedge Form(y) \wedge Var(x) \cap Var(y) = \emptyset \wedge \\ &\quad (\exists \pi)|\pi| \leq |z| \wedge \mathrm{Prf}_P(\pi, x \vee y) \wedge \neg \mathrm{Taut}(x)\end{aligned}$$

and

$$\begin{aligned}\chi(x,y,z) &= Form(x) \wedge Form(y) \wedge Var(x) \cap Var(y) = \emptyset \wedge \\ &\quad \neg \mathrm{Taut}(y)\end{aligned}$$

These formulas are straightforward first-order formalizations of $U_1(P)$ and U_2, respectively. As θ and χ are Σ^b_1-formulas they represent the sets $U_1(P)$ and U_2.

We have to verify that

$$T \vdash (\forall x)(\forall y)(\forall z)\neg\theta(x,y,z) \vee \neg\chi(x,y,z) \ .$$

For this let M be a model of T and let $(\varphi, \psi, 1^m)$ be a triple of elements from M such that $M \models \theta(\varphi, \psi, 1^m)$. Then in the model M there exists a proof π of $\varphi \vee \psi$. Because $T \vdash \mathrm{RFN}(P)$ we get $M \models \mathrm{Taut}(\varphi \vee \psi)$. As $M \models S^1_2$ we get

$$M \models Var(\varphi) \cap Var(\psi) = \emptyset \wedge \mathrm{Taut}(\varphi \vee \psi) \rightarrow \mathrm{Taut}(\varphi) \vee \mathrm{Taut}(\psi) \ .$$

But since $M \models \theta(\varphi, \psi, 1^m)$ we also have $M \models \neg\mathrm{Taut}(\varphi)$ and hence $M \models \mathrm{Taut}(\psi)$. This implies $M \models \neg\chi(\varphi, \psi, 1^m)$.

The representability of $(I_1(P), I_2(P))$ in P is shown analogously. □

Combining this lemma with the $\leq_s$-hardness of $(U_1(P), U_2)$ for $\mathsf{DNPP}(P)$ as shown in Theorem 4.8.3 we obtain:

Theorem 4.8.6 *Let $P \geq EF$ be a regular proof system that is closed under substitutions by constants. Then $(U_1(P), U_2)$ is $\leq_s$-complete for $\mathsf{DNPP}(P)$.*

For strongly regular systems P we can additionally show the $\leq_s$-completeness of the interpolation pair for $\mathsf{DNPP}(P)$:

Theorem 4.8.7 *Let $P \geq EF$ be a strongly regular proof system that is efficiently closed under substitutions by constants. Then $(U_1(P), U_2)$ and $(I_1(P), I_2(P))$ are $\leq_s$-complete for $\mathsf{DNPP}(P)$. In particular we have*

$$(U_1(P), U_2) \equiv_s (I_1(P), I_2(P)) \ .$$

Proof. The $\leq_s$-completeness of $(U_1(P), U_2)$ was already stated in Theorem 4.8.6.

As by Lemma 4.8.5 also $(I_1(P), I_2(P))$ is representable in P it remains to show that $(I_1(P), I_2(P))$ is $\leq_s$-hard for $\mathsf{DNPP}(P)$. For this let (A, B) be a disjoint NP-pair that is representable in P. By Theorem 4.5.8 we know that (A, B) is also representable in the theory T corresponding to P. Let $\varphi(x)$ and $\psi(x)$ be representations of A and B, respectively, such that

$$T \vdash (\forall x)\neg\varphi(x) \vee \neg\psi(x) \ .$$

Because P is strongly regular there exists a polynomial time computable function f that on input 1^n produces a P-proof of

$$\|\neg\varphi(x) \vee \neg\psi(x)\|^n \ .$$

Further, because by assumption P is efficiently closed under substitutions by constants we can use f to obtain a polynomial time computable function g that on input $\bar{a} \in \{0,1\}^n$ outputs a P-proof of

$$\|\neg\varphi(x) \vee \neg\psi(x)\|^n(\bar{p}^x/\bar{a}) \ .$$

We claim that the $\leq_s$-reduction from (A, B) to $(I_1(P), I_2(P))$ is given by

$$a \mapsto (\|\neg\varphi(x)\|^{|a|}(\bar{p}^x/\bar{a}), \|\neg\psi(x)\|^{|a|}(\bar{p}^x/\bar{a}), g(\bar{a}))$$

where the auxiliary variables of $\|\neg\varphi(x)\|^{|a|}$ and $\|\neg\psi(x)\|^{|a|}$ are chosen disjoint. Verifying the correctness of the reduction proceeds as in Theorem 4.8.3. □

As a corollary we get from Proposition 4.6.1 and Theorem 4.8.7 for our standard examples for strong proof systems:

Corollary 4.8.8 *Let Φ be a polynomial time set of true Π_1^b-formulas. Then for every disjoint* NP*-pair (A, B) we have*

$$(A, B) \in \mathsf{DNPP}(EF + \|\Phi\|) \iff (A, B) \leq_s (U_1(EF + \|\Phi\|), U_2) \ .$$

Additionally, we have

$$(U_1(EF + \|\Phi\|), U_2) \equiv_s (I_1(EF + \|\Phi\|), I_2(EF + \|\Phi\|)) \ .$$

The equivalence of the interpolation pair and the U-pair for strong systems as stated in the last corollary might come unexpected as the first idea for a reduction from the U-pair to the I-pair probably is to generate proofs for $\varphi \vee \psi$ at input $(\varphi, \psi, 1^m)$. This, however, is not possible for extensions of EF, because a reduction from $(U_1(P), U_2)$ to $(I_1(P), I_2(P))$ of the form

$$(\varphi, \psi, 1^m) \mapsto (\varphi, \psi, \pi)$$

implies the automatizability of the system P. But it is known that automatizability fails for strong systems $P \geq EF$ under cryptographic assumptions (cf. Section 5.1).

Clearly, for all proof systems $(\varphi, \psi, \pi) \mapsto (\varphi, \psi, 1^{|\pi|})$ computes a $\leq_p$-reduction from $(I_1(P), I_2(P))$ to $(U_1(P), U_2)$. For weak systems like resolution or cutting planes the opposite reduction is not possible unless the system is weakly automatizable. This is the content of the next proposition.

Proposition 4.8.9 *Let P be a proof system that has the feasible interpolation property and is closed under disjunctions. Then $(U_1(P), U_2) \leq_p (I_1(P), I_2(P))$ implies that P is weakly automatizable.*

Proof. By Theorem 4.4.8 feasible interpolation for P means that the interpolation pair is p-separable. Therefore $(U_1(P), U_2) \leq_p (I_1(P), I_2(P))$ implies that also $(U_1(P), U_2)$ is p-separable. Closure of P under disjunctions together with Proposition 4.8.1 guarantees

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (U_1(P), U_2) \ ,$$

hence also $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ is p-separable and therefore P is weakly automatizable. □

Of course we can use part 2 of Proposition 4.8.1 together with an analogous argument as above to infer that weak automatizability of P is also a sufficient condition to reduce $(U_1(P), U_2)$ to $(I_1(P), I_2(P))$. Instead we just state the reduction for automatizable proof systems.

Proposition 4.8.10 *Let P be an automatizable proof system. Then*

$$(U_1(P), U_2) \leq_p (I_1(P), I_2(P)) \ .$$

Proof. Let P be automatizable. Hence there exists a polynomial time computable function f that on input $(\varphi, 1^m)$ produces a P-proof of φ provided $(\varphi, 1^m) \in \mathrm{Ref}(P)$. If $(\varphi, 1^m) \notin \mathrm{Ref}(P)$ the behaviour of f is unspecified. The desired reduction is given by

$$(\varphi, \psi, 1^m) \mapsto \begin{cases} (\varphi, \psi, f(\varphi \vee \psi, 1^m)) & \text{if } P(f(\varphi \vee \psi, 1^m)) = \varphi \vee \psi \\ (\varphi_0, \psi_0, \pi_0) & \text{otherwise} \end{cases}$$

where $(\varphi_0, \psi_0, \pi_0)$ is a fixed triple from $I_2(P)$. □

4.9 Canonical $\leq_s$-Complete Pairs

The definition of the canonical pair $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ was motivated by the reflection principle (cf. Sect. 4.7). Additionally, the canonical pair is tightly connected to the automatizability of the proof system (cf. Sect. 4.4.2). In the same way the interpolation pair captures the feasible interpolation property (cf. Sect. 4.4.3). It is therefore natural to ask what is the meaning of the U-pair which we introduced in the previous section. We will argue that the U-pair is in fact the natural choice for a $\leq_s$-complete pair for the classes $\mathsf{DNPP}(P)$.

Complexity classes are usually defined by a machine model to which resource bounds are imposed. A complexity class is syntactic if the machines can be appropriately standardized such that there exists an easy test which verifies that all these standardized machines define indeed languages from the complexity class (cf. (Pap94)). For syntactic classes there is a canonical way how to define complete languages. Namely, if $\mathcal{M}$ denotes the set of all standardized machines with implicit resource bounds, then

$$\{(M, x) \mid M \in \mathcal{M} \text{ and } M(x) \text{ accepts}\}$$

is complete for the respective complexity class. For example the syntactic class NP has the following canonical $\leq_m^p$-complete language

$$\{(M, x, 1^m) \quad | \quad M \text{ is a nondeterministic Turing machine that accepts } x \text{ in } \leq m \text{ steps}\} \ .$$

The machine model for disjoint NP-pairs consists of pairs of nondeterministic polynomial time bounded Turing machines that do not accept any element

in common. This, however, is not a syntactic definition as we cannot test whether two given nondeterministic Turing machines indeed accept disjoint languages. In fact, by the theorem of Rice (Ric53) the set

$$\{(M_1, M_2) \mid M_1 \text{ and } M_2 \text{ are nondeterministic Turing machines such that } L(M_1) \cap L(M_2) = \emptyset\}$$

is undecidable. Therefore, constructing complete disjoint NP-pairs via the above method fails.

If we restrict the class of all DNPP to those disjoint NP-pairs that are representable in some fixed proof system P, then the situation is different. The machine model now consists of pairs (M_1, M_2) of polynomial time nondeterministic Turing machines such that the disjointness of $L(M_1)$ and $L(M_2)$ has polynomial size P-proofs for suitable propositional descriptions of M_1 and M_2. These propositional descriptions are computable in polynomial time from the machines M_1 and M_2. As further the polynomial size P-proofs of $L(M_1) \cap L(M_2) = \emptyset$ can be guessed and verified in polynomial time the process of checking that (M_1, M_2) defines a pair in $\mathsf{DNPP}(P)$ can be performed in nondeterministic polynomial time. Hence $\mathsf{DNPP}(P)$ is a syntactic class with hard languages defined in the canonical way. Translating this canonical hard language to the propositional level we arrive at a pair $(W_1(P), W_2(P))$ with

$$\begin{aligned} W_1(P) = \{(\varphi(\bar{x},\bar{y}), \psi(\bar{x},\bar{z}), a, 1^m) \mid\ & \mathrm{Var}(\varphi) \cap \mathrm{Var}(\psi) = \{\bar{x}\}, \\ & \varphi(\bar{a},\bar{y}) \in \mathrm{SAT} \text{ and} \\ & P \vdash_{\leq m} \neg\varphi(\bar{x},\bar{y}) \vee \neg\psi(\bar{x},\bar{z})\} \\ W_2(P) = \{(\varphi(\bar{x},\bar{y}), \psi(\bar{x},\bar{z}), a, 1^m) \mid\ & \mathrm{Var}(\varphi) \cap \mathrm{Var}(\psi) = \{\bar{x}\}, \\ & \psi(\bar{a},\bar{z}) \in \mathrm{SAT} \text{ and} \\ & P \vdash_{\leq m} \neg\varphi(\bar{x},\bar{y}) \vee \neg\psi(\bar{x},\bar{z})\} \ . \end{aligned}$$

In the components $W_1(P)$ and $W_2(P)$ the propositional formulas $\varphi(\bar{x},\bar{y})$ and $\psi(\bar{x},\bar{z})$ describe the Turing machines M_1 and M_2 for inputs of length $|\bar{x}|$. The variables $\bar{x}$ are reserved for the input whereas the variables $\bar{y}$ and $\bar{z}$ take the witness and auxiliary information necessary for the computation of the machines M_1 and M_2. The P-proofs of length $\leq m$ certify the disjointness of $L(M_1)$ and $L(M_2)$. Finally, the satisfiability conditions on $\varphi(\bar{a},\bar{y})$ and $\psi(\bar{a},\bar{z})$ describe that M_1 and M_2, respectively, accept the input a.

The W-pair and the U-pair are very similar. The essential difference is that in the U-pair the input is already substituted into the formulas describing the machines. This makes the definition of the pair somewhat simpler and displays the similarity of the pair to the interpolation pair. On the other

hand closure of the proof system P under substitutions by constants is no more necessary to show the hardness of $(W_1(P), W_2(P))$ for $\mathsf{DNPP}(P)$. However, like in the case of the canonical pair and the U-pair it is not clear that the W-pair itself is representable in P, unless the system P is regular. We collect these observations in the next theorem.

Theorem 4.9.1 *1. For any proof system P the pair $(W_1(P), W_2(P))$ is $\leq_s$-hard for the class $\mathsf{DNPP}(P)$.*

2. For any regular proof system P the pair $(W_1(P), W_2(P))$ is $\leq_s$-complete for $\mathsf{DNPP}(P)$.

4.10 Symmetry of Disjoint NP-Pairs

An interesting property of disjoint NP-pairs is the symmetry as defined by Pudlák:

Definition 4.10.1 (Pudlák (Pud03)) *A disjoint NP-pair is* symmetric *if $(A, B) \leq_p (B, A)$.*

Apparently, the symmetry of a pair (A, B) implies that $(A, B) \equiv_p (B, A)$. Symmetry of a pair means that both components look very similar, hence the pair can be given a robust definition. It is clear that all p-separable pairs are symmetric. Therefore Glaßer et al. (GSSZ04) suggest to search for non-symmetric pairs in order to establish the existence of p-inseparable pairs. One result in this direction is the following:

Theorem 4.10.2 (Glaßer, Selman, Sengupta, Zhang (GSSZ04)) *If $\mathsf{E} \neq \mathsf{NE} \cap \mathsf{coNE}$, then there is a set $A \in \mathsf{NP} \cap \mathsf{coNP}$ such that $(A, \bar{A})$ is not symmetric.*

If we look at the property of symmetry of pairs under the other reductions, then different pictures emerge. For the strong reduction $\leq_s$ it is clear that a DNPP (A, B) cannot be symmetric if we choose A from P and B NP-complete. In other words:

Proposition 4.10.3 *$\mathsf{P} \neq \mathsf{NP}$ if and only if there exist non-symmetric pairs with respect to $\leq_s$.*

A similar result for $\leq_p$ is not known as $\leq_p$-non-symmetric pairs are p-inseparable and it is not clear how to derive the existence of p-inseparable pairs from the assumption $\mathsf{P} \neq \mathsf{NP}$.

In contrast, under the Turing reduction $\leq_T$ all disjoint NP-pairs are symmetric:

Proposition 4.10.4 *All disjoint* NP*-pairs are symmetric with respect to Turing and smart Turing reductions.*

Proof. Let the pair (A, B) be separated by the set S. Then $\bar{S}$ is a separator for (B, A). The set S is Turing reducible to its complement by asking the input string as an oracle query and negating the answer. Hence the reduction is even a smart Turing reduction. □

The NP-pairs associated with propositional proof systems are usually symmetric. For the interpolation pair this is already apparent from its definition. For the canonical pair and the U-pair we get symmetry at least for sufficiently strong proof systems.

Proposition 4.10.5 *Let Φ be a polynomial time set of true Π_1^b-formulas. Then the canonical pair of $EF+\|\Phi\|$ is symmetric with respect to $\leq_p$. Further $(U_1(EF + \|\Phi\|), U_2)$ is symmetric with respect to $\leq_s$.*

Proof. Let Φ be a polynomial time set of true Π_1^b-formulas and let P denote the system $EF + \|\Phi\|$. By Corollary 4.7.3 and Theorem 4.8.6 the canonical pair and the U-pair of P are $\leq_p$- and $\leq_s$-complete for $\mathsf{DNPP}(P)$, respectively. Clearly the notion of representability is symmetric, hence $(\mathrm{SAT}^*, \mathrm{Ref}(P))$ and $(U_2, U_1(P))$ are contained in $\mathsf{DNPP}(P)$. Therefore these pairs reduce to $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ and $(U_1(P), U_2)$, respectively. □

The first part of Proposition 4.10.5 also holds for the Frege system and its bounded depths versions as already remarked by Razborov (Raz94). The decisive property for the symmetry of the canonical pair and the U-pair is the reflection property of the proof system. But also weaker systems without reflection have symmetric pairs. For resolution this was shown by Pudlák (Pud03).

4.11 NP-Pairs and the Simulation Order of Proof Systems

Now we use the results of the last sections to make some observations about the connection between the simulation order of proof systems and disjoint NP-pairs. As this analysis frequently involves proof systems with suitable closure properties which we want to avoid to list at each occasion we make the following definition:

Definition 4.11.1 *We call a proof system P* strong *if $P \geq EF$ is a regular proof system that is closed under modus ponens, disjunctions and substitutions by constants.*

For instance, all extensions of EF by translations of true arithmetic formulas are strong in this sense, and therefore every proof system is simulated by some strong system. If we are interested in exploring optimal proof systems, then it is anyway legitimate to make as many assumptions on the systems as necessary:

Proposition 4.11.2 *If P is an optimal proof system, then P is strong.*

Proof. By Proposition 3.7.5 we have $P \leq_p EF + \|\mathrm{RFN}(P)\|$ and by the optimality of P also $EF + \|\mathrm{RFN}(P)\| \leq P$. Hence the systems P and $EF + \|\mathrm{RFN}(P)\|$ are $\leq$-equivalent. Therefore, by Proposition 2.6.5 the system P has all the required closure properties. The sequence $\|\mathrm{RFN}(P)\|^n$ is polynomial time constructible and hence has polynomial size P-proofs. This means that P has reflection and therefore by Theorem 3.7.1 the system P is regular. □

We start our analysis with an easy but very useful observation from (Pud03) expressing that the simulation order of propositional proof systems is reflected in reductions between the canonical pairs.

Proposition 4.11.3 (Pudlák (Pud03)) *If P and S are proof systems with $P \leq S$, then we have*

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(S), \mathrm{SAT}^*) \ .$$

Proof. By assumption there is a polynomial p, such that for all formulas φ and P-proofs π of φ there is a S-proof π' of length $\leq p(|\pi|)$. Therefore the mapping

$$(\varphi, 1^m) \mapsto (\varphi, 1^{p(m)})$$

is a $\leq_p$-reduction from $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ to $(\mathrm{Ref}(S), \mathrm{SAT}^*)$. □

Probably not unexpected, this link between simulations of propositional proof systems and reductions between disjoint NP-pairs extends to the question of the existence of maximal elements in the respective orders. The following theorem which is usually attributed to Razborov (Raz94) expresses this for the reduction $\leq_p$. Actually, the result as such is not stated in (Raz94), but it easily follows from the results proven there.

Theorem 4.11.4 *If P is an optimal proof system, then the canonical pair of P is a $\leq_p$-complete disjoint NP-pair.*

Proof. Let the proof system P be optimal and let (A, B) be some disjoint NP-pair. We choose arbitrary representations φ_n and ψ_n for A and B, respectively. Now we construct some strong proof system that admits polynomial size proofs of $\neg\varphi_n \vee \neg\psi_n$. For example,

$$Q = EF + \{\neg\varphi_n \vee \neg\psi_n \mid n \geq 0\}$$

is such a proof system. By Theorem 4.6.3 we get

$$(A, B) \leq_p (\mathrm{Ref}(Q), \mathrm{SAT}^*) \ .$$

Because P is optimal we have $Q \leq P$ and hence by Proposition 4.11.3 we get

$$(\mathrm{Ref}(Q), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(P), \mathrm{SAT}^*) \ .$$

Combining these reductions we get the reduction from (A, B) to the canonical pair of P as claimed. □

Even without assuming the existence of optimal proof systems we can say that candidates for $\leq_p$-complete NP-pairs come from canonical pairs of strong proof systems:

Proposition 4.11.5 *Let (A, B) be $\leq_p$-complete for the class of all DNPP. Then we have $(A, B) \equiv_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$ for some strong proof system P.*

Proof. As in the last proof we choose some strong proof system Q such that (A, B) is representable in Q. Then $(A, B) \leq_p (\mathrm{Ref}(Q), \mathrm{SAT}^*)$ and by assumption $(\mathrm{Ref}(Q), \mathrm{SAT}^*) \leq_p (A, B)$. □

We now analyse how the simulation order of proof systems is reflected in the more refined reduction $\leq_s$. In Sect. 4.1 it was shown that the reductions $\leq_p$ and $\leq_s$ are different under the assumption $\mathsf{P} \neq \mathsf{NP}$. Still we have:

Proposition 4.11.6 *Let P be a strong proof system. Then for all disjoint NP-pairs (A, B) it holds*

$$(A, B) \leq_p (U_1(P), U_2) \quad \Longleftrightarrow \quad (A, B) \leq_s (U_1(P), U_2) \ .$$

Proof. Let $(A, B) \leq_p (U_1(P), U_2)$. By Lemma 4.8.5 $(U_1(P), U_2)$ is representable in P. Hence with Proposition 4.6.1 also (A, B) is representable in P, from which we conclude with Theorem 4.8.3

$$(A, B) \leq_s (U_1(P), U_2) \ .$$

The opposite implication holds by definition. □

Corollary 4.11.7 *Let P and S be strong proof systems. Then we have:*

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(S), \mathrm{SAT}^*) \iff (U_1(P), U_2) \leq_s (U_1(S), U_2) \ .$$

Proof. For the first direction we get from

$$(U_1(P), U_2) \leq_p (\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(S), \mathrm{SAT}^*) \leq_p (U_1(S), U_2)$$

together with the last proposition

$$(U_1(P), U_2) \leq_s (U_1(S), U_2) \ .$$

The other implication follows from

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (U_1(P), U_2) \leq_p (U_1(S), U_2) \leq_p (\mathrm{Ref}(S), \mathrm{SAT}^*) \ .$$

□

Proposition 4.11.3 and Corollary 4.11.7 yield an analogon of Proposition 4.11.3 for strong proof systems:

Corollary 4.11.8 *If P and S are strong proof systems with $P \leq S$, then we have*

$$(U_1(P), U_2) \leq_s (U_1(S), U_2) \ .$$

Köbler, Messner and Torán proved in (KMT03) that the existence of an optimal proof system implies the existence of $\leq_s$-complete NP-pairs. This result also follows from our observations here. Additionally, we can exhibit a complete pair in this case:

Theorem 4.11.9 *If P is an optimal proof system, then $(U_1(P), U_2)$ is $\leq_s$-complete for the class of all DNPP.*

Proof. Let P be an optimal proof system and (A, B) a DNPP. We choose arbitrary propositional representations φ_n and ψ_n for A and B, respectively. As the sequence $\neg\varphi_n \vee \neg\psi_n$ is constructible in polynomial time there exists some proof system with polynomial size proofs of these tautologies. Because P is optimal we also have polynomial size P-proofs of $\neg\varphi_n \vee \neg\psi_n$, hence (A, B) is representable in P. The system P is optimal, so in particular it is strong by Proposition 4.11.2. Therefore we can apply Theorem 4.8.6 to conclude $(A, B) \leq_s (U_1(P), U_2)$.

Therefore the pair $(U_1(P), U_2)$ is $\leq_s$-complete for all DNPP. □

In the same way as Proposition 4.11.5 we get:

Proposition 4.11.10 *Let (A, B) be $\leq_s$-complete for the class of all DNPP. Then we have $(A, B) \equiv_s (U_1(P), U_2)$ for some strong proof system P.*

We now turn again to the question whether complete pairs exists, but without assuming the existence of optimal proof systems. Glaßer, Selman and Sengupta (GSS05) proved that the answer to the problem does not depend on the strength of the reductions used. In (GSS05) the following result is proved by elementary, but involved simulation techniques. Here we give an easy proof based on our results from this chapter.

Theorem 4.11.11 (Glaßer, Selman, Sengupta (GSS05)) *The class of all disjoint* NP*-pairs contains a $\leq_p$-complete pair if and only if it contains a $\leq_s$-complete pair.*

Proof. For the first direction we can assume with Proposition 4.11.5 that the $\leq_p$-complete DNPP has the form $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ for some strong proof system P. Then all disjoint NP-pairs are representable in P and therefore by Theorem 4.8.3 all DNPP are $\leq_s$-reducible to $(U_1(P), U_2)$.

The other direction holds by definition. □

In (GSS05) Glaßer et al. prove that the existence of a complete DNPP under smart Turing reductions already implies the existence of a $\leq_p$-complete DNPP (and hence by Theorem 4.11.11 also of a $\leq_s$-complete pair). We can easily reprove their result in our framework by noticing:

Lemma 4.11.12 *Let $T \supseteq S_2^1$ be an L-theory. Then the class* DNPP(T) *is closed under smart Turing reductions.*

Proof. Let the pair (A, B) be smartly Turing reducible to (C, D) via the deterministic oracle Turing machine M, and let (C, D) be representable in T. Consider the NP-sets

$$\begin{aligned} A' &= \{x \mid x \in A \text{ and } M(x) \text{ accepts}\} \\ B' &= \{x \mid x \in B \text{ and } M(x) \text{ rejects}\} \ . \end{aligned}$$

By "$M(x)$ accepts" we mean that M accepts the input x by a computation where all oracle queries that are positively answered are verified by a computation of a nondeterministic machine for C and all negative answers are verified by D. Since the reduction is smart we have $A = A'$ and $B = B'$. For $T \vdash A' \cap B' = \emptyset$ it suffices to show in T the uniqueness of the computation of M at inputs x from $A \cup B$. Because T is an extension of S_2^1 it can prove the uniqueness of computations of the deterministic machine M, and the possibility to answer an oracle query both positively and negatively is excluded by $T \vdash C \cap D = \emptyset$. □

From this we conclude:

Proposition 4.11.13 *Suppose (A, B) is a smart $\leq_T$-complete pair. Let $T \supseteq S_2^1$ be an arithmetic theory in which (A, B) is representable. Then the pair $(U_1(P), U_2)$ is $\leq_s$-complete for all DNPP where P is the proof system corresponding to T.*

Proof. We choose arithmetic representations φ and ψ of A and B, respectively, and define the theory T as $S_2^1 + \neg\varphi \vee \neg\psi$. Then by the last lemma all DNPP are representable in T. By Theorem 4.5.8 this implies that all pairs are representable in the proof system $P = EF + \|\neg\varphi \vee \neg\psi\|$ and therefore the pair $(U_1(P), U_2)$ is $\leq_s$-complete by Theorem 4.8.3. □

It is not clear whether the class of pairs representable in some theory T is also closed under $\leq_T$-reductions. This corresponds to the open problem from (GSS05) whether the existence of a $\leq_T$-complete pair implies the existence of a $\leq_p$-complete DNPP.

We will continue the investigation of complete NP-pairs in Sect. 6.5 where we provide further characterizations for their existence in the more general context of disjoint k-tuples of NP-sets.

4.12 A Weak Reduction Between Proof Systems

This section is devoted to the analysis of a weak notion of simulation for proof systems introduced in (KP89) but not much studied elsewhere. This simulation is provably weaker than the ordinary reduction between proof systems but is equivalent with respect to the existence of optimal proof systems. In the next section we will relate the simulation order of proof systems under this weaker reduction with the reductions between canonical pairs.

The reduction is defined as follows:

Definition 4.12.1 (Krajíček, Pudlák (KP89)) *Let P and Q be propositional proof systems. Then $P \leq' Q$ holds if for all polynomials p there exists a polynomial q such that*

$$P \vdash_{\leq p(|\varphi|)} \varphi \quad \textit{implies} \quad Q \vdash_{\leq q(|\varphi|)} \varphi$$

for all tautologies φ.

Using the notation $\vdash_*$ which hides the actual polynomials we can also express the reduction $\leq'$ more compactly as: $P \leq' Q$ if and only if for all sets Φ of tautologies

$$P \vdash_* \Phi \quad \text{implies} \quad Q \vdash_* \Phi \ .$$

Let us try to motivate the above definition. If we express combinatorial principles in propositional logic or if we translate true arithmetic formulas into propositional formulas we arrive at collections Φ of tautologies that typically contain one tautology per input length. We say that a proof system P proves a combinatorial principle or an arithmetic formula if there exist polynomially long P-proofs of the corresponding collection of tautologies. If $P \leq Q$, then every principle that is provable in P is also provable in Q. The Q-proofs are allowed to be longer than the P-proofs but only up to fixed polynomial amount independent of the principle proven. The reduction $\leq'$ is more flexible as it allows a different polynomial increase for each principle.

To prove $P \not\leq Q$ one typically shows super-polynomial lower bounds on the length of Q-proofs of some principle like e.g. the pigeon hole principle whereas the principle is provable in P. As basically all separations between proof systems are achieved in this manner all these results also separate the corresponding proof systems with respect to the weaker $\leq'$-reduction.

To further motivate the definition we remark that we can characterize an ordinary $\leq$-simulation of P by Q by

$$(\exists q \in \text{Poly})(\forall p \in \text{Poly})(\forall \varphi)\ P \vdash_{\leq p(|\varphi|)} \varphi \implies Q \vdash_{\leq q(p(|\varphi|))} \varphi$$

where Poly denotes the set of all polynomials. On the other hand it is easily seen that $P \leq' Q$ holds if and only if

$$(\forall p \in \text{Poly})(\exists q \in \text{Poly})(\forall \varphi)\ P \vdash_{\leq p(|\varphi|)} \varphi \implies Q \vdash_{\leq q(p(|\varphi|))} \varphi \ .$$

Hence we get the definition of $\leq'$ by changing the order of the quantifiers from $\exists q \forall p$ to $\forall p \exists q$ in the above characterization of $\leq$.

It is clear from the above explanation that $\leq$ is a refinement of $\leq'$. We first observe that it is indeed a proper refinement, i.e. we can separate $\leq$ and $\leq'$. It is, however, not possible to achieve this separation with regular proof systems.

Proposition 4.12.2 *1. Let P be a proof system that is not polynomially bounded. Then there exists a proof system Q such that $P \leq' Q$ but $P \not\leq Q$.*

2. Let Φ and Ψ be polynomial time sets of tautologies. Then $EF + \Phi \leq' EF + \Psi$ implies $EF + \Phi \leq EF + \Psi$.

Proof. To prove part 1 let P be a proof system that is not polynomially bounded. We define the system Q. Q-proofs consist of multiple copies of P-proofs where the number of copies depends on the length of the P-proof, more precisely $Q(\pi) = \varphi$ if there exists a P-proof π' of φ such that $\pi = (\pi')^l$ where the number l of the copies of π' is determined as follows. Let k be a number such that $|\varphi|^{k-1} \leq |\pi'| < |\varphi|^k$. Then l is chosen as $l = |\varphi|^{(k-1)k}$. Hence we have

$$|\varphi|^{k-1}|\varphi|^{(k-1)k} = |\varphi|^{k^2-1} \leq |\pi| < |\varphi|^k|\varphi|^{(k-1)k} = |\varphi|^{k^2} \ .$$

P is $\leq'$-simulated by Q because for each polynomial p majorized by n^k we can choose q as n^{k^2}, i.e.

$$P \vdash_{\leq|\varphi|^k} \varphi \implies Q \vdash_{\leq|\varphi|^{k^2}} \varphi \ .$$

But if P is not polynomially bounded, then there is apparently no polynomial q such that

$$P \vdash_{\leq m} \varphi \implies Q \vdash_{\leq q(m)} \varphi \ ,$$

i.e. $P \not\leq Q$.

Now we prove part 2. Let Φ and Ψ be polynomial time sets of tautologies. Let us denote the systems $EF + \Phi$ and $EF + \Psi$ by P and Q, respectively. The regularity of P implies $P \vdash_* \|\mathrm{RFN}(P)\|^n$. Because $P \leq' Q$ we also have $Q \vdash_* \|\mathrm{RFN}(P)\|^n$. Using Lemma 3.7.6 we infer $P \leq Q$ as claimed. □

We call a proof system $\leq'$-optimal if it $\leq'$-simulates all proof systems. Krajíček and Pudlák (KP89) proved that the existence of a $\leq'$-optimal proof system already implies the existence of an optimal proof system. Comparing $\leq$ and $\leq_p$ it is interesting to mention that it is neither known how to separate these reductions nor how to infer from the existence of an optimal proof system the existence of a p-optimal proof system.

Theorem 4.12.3 (Krajíček, Pudlák (KP89)) *There exists an optimal proof system if and only if there exists a $\leq'$-optimal proof system.*

Proof. The forward direction is immediate as $\leq$ is a refinement of $\leq'$.

For the reverse implication let P be a $\leq'$-optimal proof system. We claim that the proof system

$$P' = EF + \|\mathrm{RFN}(P)\|$$

is optimal. To see this let Q be a proof system. Consider the proof system $Q' = EF + \|\mathrm{RFN}(Q)\|$. Obviously $Q' \vdash_* \|\mathrm{RFN}(Q)\|^n$. Because P is $\leq'$-optimal we have $Q' \leq' P$ and hence $P \vdash_* \|\mathrm{RFN}(Q)\|^n$. From the definition of P' and Proposition 3.7.5 we get $P \leq_p P'$ and therefore also

$P' \vdash_* \|\mathrm{RFN}(Q)\|^n$. Since P' is regular we infer with Lemma 3.7.6 $Q \leq P'$ as desired. □

As we already know that the existence of optimal proof systems implies the existence of complete DNPP we can formulate the following corollary:

Corollary 4.12.4 *If there exists a $\leq'$-optimal proof system, then there exist disjoint* NP*-pairs which are $\leq_p$- and $\leq_s$-complete for the class of all DNPP.*

4.13 Proof Systems with Equivalent Canonical Pairs

Already in Sect. 4.11 we have used the close relation between the simulation order of proof systems and the reductions between canonical pairs. Essentially, this connection rests upon the fact that $\mathsf{DNPP}(P)$ is a subclass of $\mathsf{DNPP}(Q)$ if the proof systems P is simulated by the system Q. For the canonical pairs this is expressed by the following observation (already stated earlier as Proposition 4.11.3):

Proposition 4.13.1 (Pudlák (Pud03)) *If P and Q are proof systems with $P \leq Q$, then the canonical pair of P is $\leq_p$-reducible to the canonical pair of Q.*

Proof. The reduction is given by $(\varphi, 1^m) \mapsto (\varphi, 1^{p(m)})$ where p is the polynomial from $P \leq Q$. □

We will now explore how tight the connection between the simulation order of proof systems and reductions in the lattice of pairs really is, i.e. to what extend the opposite implication of Proposition 4.13.1 is valid. If $P \not\leq Q$, then we cannot hope to reduce $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ to $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ by a reduction of the form $(\varphi, 1^m) \mapsto (\varphi, 1^n)$ that changes only the proof length but leaves the formula unchanged. However, unlike in the case of simulations between proof systems the reductions between canonical pairs have the flexibility to change the formula.

The aim of this section is to provide different techniques for the construction of non-equivalent proof systems with equivalent pairs. One such example is given by Pudlák in (Pud03) where he shows that two versions of the cutting planes proof system CP which do not $\leq$-simulate each other have $\leq_p$-equivalent canonical pairs. Here we search for general conditions on proof systems which imply the equivalence of the canonical pairs. The first condition will be the $\leq'$-equivalence of the proof systems. For this we show an analogue of Proposition 4.13.1 for $\leq'$.

Proposition 4.13.2 *Let P be a proof system that is closed under disjunctions and let Q be a proof system such that $P \leq' Q$. Then* $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(Q), \mathrm{SAT}^*)$.

Proof. We claim that for some suitable polynomial q the mapping

$$(\varphi, 1^m) \mapsto (\varphi \vee \bot^m, 1^{q(m)})$$

performs the desired $\leq_p$-reduction where $\bot^m$ stands for $\bot \vee \ldots \vee \bot$ (m disjuncts). To see this let first $(\varphi, 1^m) \in \mathrm{Ref}(P)$. Because P is closed under disjunctions there exists a polynomial p such that $P \vdash_{\leq m} \varphi$ implies $P \vdash_{\leq p(m)} \varphi \vee \bot^m$. Because of $P \leq' Q$ there is a polynomial q such that $Q \vdash_{\leq q(m)} \varphi \vee \bot^m$, i.e $(\varphi \vee \bot^m, 1^{q(m)}) \in \mathrm{Ref}(Q)$.

If $(\varphi, 1^m) \in \mathrm{SAT}^*$, then the satisfiability of $\neg\varphi$ is transferred to $\neg(\varphi \vee \bot^m) = \neg\varphi \wedge \top \wedge \ldots \wedge \top$. □

Combining Propositions 4.12.2 and 4.13.2 we get the afore mentioned counterexamples to the converse of Proposition 4.13.1.

Corollary 4.13.3 *Let P be a proof system that is closed under disjunctions and is not polynomially bounded. Then there exists a proof system Q such that*

$$P \not\equiv Q \quad \textit{and} \quad (\mathrm{Ref}(P), \mathrm{SAT}^*) \equiv_p (\mathrm{Ref}(Q), \mathrm{SAT}^*) \ .$$

Proof. The proof system Q constructed from P in Proposition 4.12.2 fulfills $P \leq' Q \leq P$ and $P \not\leq Q$. Hence $P \not\equiv Q$.

By Proposition 4.13.1 we have $(\mathrm{Ref}(Q), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$ and applying Proposition 4.13.2 we conclude $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(Q), \mathrm{SAT}^*)$. □

The proof systems P and Q from the last corollary have equivalent canonical pairs and are also $\leq'$-equivalent. Moreover it follows from Proposition 4.13.2 that the canonical pair of a disjunctively closed proof system is already determined by the $\leq'$-degree of the system. More precisely:

Proposition 4.13.4 *Let P and Q be $\leq'$-equivalent proof systems that are closed under disjunctions. Then* $(\mathrm{Ref}(P), \mathrm{SAT}^*) \equiv_p (\mathrm{Ref}(Q), \mathrm{SAT}^*)$.

Nevertheless we can also construct proof systems that have equivalent canonical pairs but are not $\leq'$-equivalent. We show this in the next proposition.

Proposition 4.13.5 *Let P be a proof system such that the system $EF + \|\mathrm{RFN}(P)\|$ is not optimal. Then there exists a proof system Q such that*

$$Q \not\equiv' P \quad \textit{and} \quad (\mathrm{Ref}(P), \mathrm{SAT}^*) \equiv_p (\mathrm{Ref}(Q), \mathrm{SAT}^*) \ .$$

Proof. Because $EF+\|\mathrm{RFN}(P)\|$ is not optimal there exists by Theorem 3.7.8 a sequence of polynomial time constructible tautologies φ_n such that

$$EF + \|\mathrm{RFN}(P)\| \not\vdash_* \varphi_n \ .$$

As P is simulated by $EF + \|\mathrm{RFN}(P)\|$ the sequence φ_n is also hard for P, i.e. $P \not\vdash_* \varphi_n$. We define Q as

$$Q(\pi) = \begin{cases} P(\pi') & \text{if } \pi = 0\pi' \\ \varphi_n & \text{if } \pi = 1\varphi_n \text{ for some } n \\ \top & \text{otherwise.} \end{cases}$$

Clearly, $P \leq Q$ and therefore $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(Q), \mathrm{SAT}^*)$. The converse reduction from $(\mathrm{Ref}(Q), \mathrm{SAT}^*)$ to $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ is given by

$$(\varphi, 1^m) \mapsto \begin{cases} (\psi, 1^k) & \text{if } \varphi = \varphi_n \text{ for some } n \text{ or } \varphi = \top \\ (\varphi, 1^{m-1}) & \text{otherwise} \end{cases}$$

where ψ is some fixed tautology with a P-proof of length k.

Finally, since $P \not\vdash_* \varphi_n$ and $Q \vdash_* \varphi_n$ we have $Q \not\leq' P$. □

The proof systems Q constructed in Proposition 4.13.5 have the drawback that they do not satisfy the normality conditions from Sect. 2.6. In the next theorem we will construct proof systems with somewhat better properties.

Theorem 4.13.6 *Let P be a line based proof system that allows efficient deduction and let Φ be a sparse set of tautologies which can be generated in polynomial time. Then*

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \equiv_p (\mathrm{Ref}(P \cup \Phi), \mathrm{SAT}^*) \ .$$

Proof. As P is simulated by $P \cup \Phi$ we get

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (\mathrm{Ref}(P \cup \Phi), \mathrm{SAT}^*) \ .$$

Now we describe the converse reduction. Let p be the polynomial from the efficient deduction property of P. Because Φ is a sparse set there exists a polynomial q such that for each number m the set Φ contains at most $q(m)$ tautologies of length $\leq m$. Let $\Phi_m = \Phi \cap \Sigma^{\leq m}$ be the set of these tautologies.

Then $(\mathrm{Ref}(P \cup \Phi), \mathrm{SAT}^*)$ reduces to $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ via the function

$$(\psi, 1^m) \ \mapsto \ (\ (\bigwedge_{\varphi \in \Phi_m} \varphi) \to \psi, 1^{p(mq(m)+m)}) \ .$$

To verify the claim assume that $(\psi, 1^m) \in \mathrm{Ref}(P \cup \Phi)$. Let π be a $P \cup \Phi$-proof of ψ of length $\leq m$. This proof π can use only formulas of length $\leq m$ from Φ of which there are only $\leq q(m)$ many. Hence the tautologies used in the proof π are contained in $\bigwedge_{\varphi \in \Phi_m} \varphi$. Therefore we know that π is also a proof for ψ in the proof system $P \cup \Phi_m$. Using the efficient deduction property of P we get a P-proof of size $\leq p(mq(m) + m)$ of $(\bigwedge_{\varphi \in \Phi_m} \varphi) \to \psi$.

Now assume $(\psi, 1^m) \in \mathrm{SAT}^*$. Then $\neg\psi$ is satisfiable and therefore

$$\neg((\bigwedge_{\varphi \in \Phi_m} \varphi) \to \psi) \;=\; (\bigwedge_{\varphi \in \Phi_m} \varphi) \wedge \neg\psi$$

is also satisfiable because $(\bigwedge_{\varphi \in \Phi_m} \varphi)$ is a tautology. □

By Theorem 3.7.8 we know that for any non-optimal proof system we can find a sequence of hard tautologies. Hence we get:

Corollary 4.13.7 *Let P be a line based proof system admitting efficient deduction and such that $EF + \|\mathrm{RFN}(P)\|$ is not optimal. Then there exists a sparse set Φ of tautologies which can be generated in polynomial time such that*

$$P \cup \Phi \not\leq' P \quad \textit{and} \quad (\mathrm{Ref}(P), \mathrm{SAT}^*) \equiv_p (\mathrm{Ref}(P \cup \Phi), \mathrm{SAT}^*) \ .$$

Because EF admits efficient deduction (Theorem 2.4.2) we can formulate the following corollary:

Corollary 4.13.8 *Let Φ be a sparse set of tautologies which can be generated in polynomial time. Then we have*

$$(\mathrm{Ref}(EF), \mathrm{SAT}^*) \equiv_p (\mathrm{Ref}(EF \cup \Phi), \mathrm{SAT}^*) \ .$$

As explained in Sect. 3.7 every proof system P is simulated by $EF + \|\mathrm{RFN}(P)\|$. Clearly $\|\mathrm{RFN}(P)\|$ is a sparse polynomial time set of tautologies. From this information together with Corollary 4.13.8 it might be tempting to deduce that the canonical pair of EF is $\leq_p$-complete for the class of all disjoint NP-pairs. The problem, however, is that Corollary 4.13.8 only holds for the system $EF \cup \|\mathrm{RFN}(P)\|$ whereas to show the $\leq_p$-completeness of $(\mathrm{Ref}(EF), \mathrm{SAT}^*)$ we would need it for $EF + \|\mathrm{RFN}(P)\|$. We can formulate this observation somewhat differently as:

Theorem 4.13.9 *At least one of the following is true:*

1. *The canonical pair of EF is complete for the class of all disjoint* NP*-pairs.*

2. There exists a proof system P such that

$$EF \ \leq_p \ EF \cup \|\mathrm{RFN}(P)\| \ \leq_p \ EF + \|\mathrm{RFN}(P)\|$$

is a chain of pairwise non-equivalent proof systems.

Proof. Assume that 2 fails. We will show that $(\mathrm{Ref}(EF), \mathrm{SAT}^*)$ is complete for the class of all DNPP. To prove this let (A, B) be a disjoint NP-pair. Choose some proof system P such that (A, B) is representable in P and P is closed under substitutions by constants and modus ponens and can evaluate formulas without variables. Because (A, B) is representable in P we can use Theorem 4.6.3 to infer that

$$(A, B) \ \leq_p \ (\mathrm{Ref}(P), \mathrm{SAT}^*) \ .$$

Since condition 2 fails for P we have $EF \equiv EF \cup \|\mathrm{RFN}(P)\|$ or $EF \cup \|\mathrm{RFN}(P)\| \equiv EF + \|\mathrm{RFN}(P)\|$. If $EF \equiv EF \cup \|\mathrm{RFN}(P)\|$, then $EF \vdash_* \|\mathrm{RFN}(P)\|$. By Lemma 3.7.6 this implies $P \leq EF$ and hence Proposition 4.13.1 yields

$$(A, B) \leq_p (\mathrm{Ref}(EF), \mathrm{SAT}^*) \ .$$

Now assume that $EF \cup \|\mathrm{RFN}(P)\| \equiv EF + \|\mathrm{RFN}(P)\|$ is satisfied for P. By Proposition 3.7.5

$$P \ \leq_p \ EF + \|\mathrm{RFN}(P)\|$$

and hence

$$(\mathrm{Ref}(P), \mathrm{SAT}^*) \ \leq_p \ (\mathrm{Ref}(EF + \|\mathrm{RFN}(P)\|), \mathrm{SAT}^*) \ .$$

By assumption we have

$$EF + \|\mathrm{RFN}(P)\| \ \leq \ EF \cup \|\mathrm{RFN}(P)\| \ .$$

Hence Proposition 4.13.1 and Corollary 4.13.8 give us

$$\begin{aligned} (\mathrm{Ref}(EF + \|\mathrm{RFN}(P)\|), \mathrm{SAT}^*) \ &\leq_p \ (\mathrm{Ref}(EF \cup \|\mathrm{RFN}(P)\|), \mathrm{SAT}^*) \\ &\leq_p \ (\mathrm{Ref}(EF), \mathrm{SAT}^*) \ . \end{aligned}$$

Combining all these reductions we arrive at

$$(A, B) \ \leq_p \ (\mathrm{Ref}(EF), \mathrm{SAT}^*) \ ,$$

as desired. □

Both assertions of Theorem 4.13.9 contain important information. The first alternative would solve the open problem, posed by Razborov (Raz94),

on the existence of complete pairs. But also part 2 is interesting as there is only very limited knowledge about strong proof systems $P \geq EF$.

To determine which of the alternatives from Theorem 4.13.9 is true it seems to be necessary to find out if the systems $EF \cup \Phi$ and $EF + \Phi$ can be different for some polynomial time computable set Φ of tautologies. As $EF \cup \Phi$ is closed under modus ponens this essentially means to decide whether $EF \cup \Phi$ is also closed under substitutions. However, if complete NP-pairs do not exist, then the system $EF \cup \Phi$ is not even closed under substitutions by constants for suitably chosen $\Phi \subseteq \mathrm{TAUT}$. This is the content of the next theorem.

Theorem 4.13.10 *If for all polynomial time computable sets $\Phi \subseteq \mathrm{TAUT}$ the proof system $EF \cup \Phi$ is closed under substitutions by constants, then $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ is complete for all disjoint NP-pairs.*

Proof. Let (A, B) be a disjoint NP-pair and let φ_n and ψ_n be propositional representations for A and B, respectively. Consider the proof system

$$P = EF \cup \{\neg\varphi_n \vee \neg\psi_n \mid n \geq 0\} \ .$$

Clearly, (A, B) is representable in P and hence by Theorem 4.6.3 the pair (A, B) is $\leq_p$-reducible to the canonical pair of P. By Corollary 4.13.8 this implies that (A, B) is also $\leq_p$-reducible to the canonical pair of EF. □

4.14 Different Scenarios for $\mathsf{DNPP}(P)$

In Sect. 4.6 we showed that the canonical pair of a proof system P is $\leq_p$-hard for $\mathsf{DNPP}(P)$ provided that the system P has sufficient closure properties. In the next theorem we give examples for proof systems P where the canonical pair of P is not hard for $\mathsf{DNPP}(P)$. Proving such a result requires a suitable hypothesis as $\mathsf{P} = \mathsf{NP}$ for example implies that all pairs with nonempty components are $\leq_p$-complete for the class of all DNPP. Here the assumption is that the canonical pair of EF is not $\leq_p$-complete, and this assumption even characterizes the assertion.

Theorem 4.14.1 *There exists a sparse polynomial time constructible set Φ of tautologies such that the canonical pair of $EF \cup \Phi$ is not $\leq_p$-hard for the class $\mathsf{DNPP}(EF \cup \Phi)$ if and only if $(\mathrm{Ref}(EF), \mathrm{SAT}^*)$ is not $\leq_p$-complete for all pairs.*

Proof. For the first direction assume that for some sparse polynomial time constructible set $\Phi \subseteq \mathrm{TAUT}$ the canonical pair of $EF \cup \Phi$ is not $\leq_p$-hard for $\mathsf{DNPP}(EF \cup \Phi)$. Then there exists a disjoint NP-pair (A, B) such that

$$(A, B) \not\leq_p (\mathrm{Ref}(EF \cup \Phi), \mathrm{SAT}^*) \ .$$

By Corollary 4.13.8 we know that the canonical pairs of EF and $EF \cup \Phi$ are $\leq_p$-equivalent. Therefore $(A, B) \not\leq_p (\mathrm{Ref}(EF), \mathrm{SAT}^*)$ and hence the canonical pair of EF is not $\leq_p$-complete.

For the opposite direction assume that EF is not $\leq_p$-complete. Then there exists a disjoint NP-pair (A, B) such that

$$(A, B) \not\leq_p (\mathrm{Ref}(EF), \mathrm{SAT}^*) \ .$$

We choose some representations φ_n and ψ_n of A and B, respectively, and define the system P as

$$P = EF \cup \{\neg\varphi_n \vee \neg\psi_n \mid n \geq 0\} \ .$$

By definition we have $P \vdash_* \neg\varphi_n \vee \neg\psi_n$, hence (A, B) is representable in P. By Corollary 4.13.8 we have $(\mathrm{Ref}(EF), \mathrm{SAT}^*) \equiv_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$. Hence $(A, B) \leq_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$ would imply $(A, B) \leq_p (\mathrm{Ref}(EF), \mathrm{SAT}^*)$ in contradiction to our assumption. □

In Theorem 4.6.3 we proved that the canonical pair of a proof system P is $\leq_p$-hard for $\mathsf{DNPP}(P)$ provided that P is closed under modus ponens and substitutions by constants and can evaluate formulas without variables. The counterexamples $EF \cup \Phi$ from the last theorem are closed under modus ponens and evaluate formulas without variables. Therefore the hypothesis that P is closed under substitutions by constants seems indeed to be necessary.

In the next table we summarize some of the results for the class $\mathsf{DNPP}(P)$ for some typical proof systems P. This comparison demonstrates that proof systems P with different properties give rise to different scenarios for $\mathsf{DNPP}(P)$ and the reductions between the NP-pairs associated with P.

4.15 On the Complexity of $\mathrm{Ref}(P)$

In the last table we summarized our knowledge about the reductions between the pairs associated with a proof system. One question that is left open in this connection is how $(\mathrm{Ref}(P), \mathrm{SAT}^*)$ and $(U_1(P), U_2)$ compare with respect to the strong reduction $\leq_s$. At least for regular systems with sufficient closure properties we know that $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_s (U_1(P), U_2)$. Since $U_1(P)$ is

weak systems P	resolution, cutting planes
$(\mathrm{Ref}(P), \mathrm{SAT}^*)$	$\leq_p$-hard for $\mathsf{DNPP}(P)$
$(U_1(P), U_2)$	$\leq_s$-hard for $\mathsf{DNPP}(P)$
$(I_1(P), I_2(P))$	p-separable (Pud03)
reductions	$(I_1(P), I_2(P)) \leq_p (U_1(P), U_2) \equiv_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$ $(U_1(P), U_2) \not\leq_p (I_1(P), I_2(P))$ unless P is weakly automatizable closure of $\mathsf{DNPP}(P)$ under $\leq_p$ and $\leq_s$
properties	closed under modus ponens and substitutions by constants feasible interpolation (Kra97; BPR97; Pud97) no reflection for resolution (AB02)
strong systems P	extensions $EF + \|\Phi\|$ of EF by translations of polynomial time computable sets of true Π_1^b-formulas Φ
$(\mathrm{Ref}(P), \mathrm{SAT}^*)$	$\leq_p$-complete for $\mathsf{DNPP}(P)$
$(U_1(P), U_2)$	$\leq_s$-complete for $\mathsf{DNPP}(P)$
$(I_1(P), I_2(P))$	$\leq_s$-complete for $\mathsf{DNPP}(P)$
reductions	$(I_1(P), I_2(P)) \equiv_s (U_1(P), U_2) \equiv_p (\mathrm{Ref}(P), \mathrm{SAT}^*)$ closure of $\mathsf{DNPP}(P)$ under smart $\leq_T$, $\leq_p$ and $\leq_s$
properties	closed under modus ponens and substitutions no feasible interpolation under cryptographic assumptions (KP98) strong reflection, strongly regular
other systems P	extensions $EF \cup \Phi$ of EF by suitable choices of polynomial time constructible sets $\Phi \subseteq \mathrm{TAUT}$
$(\mathrm{Ref}(P), \mathrm{SAT}^*)$	not $\leq_p$-hard for $\mathsf{DNPP}(P)$ unless $(\mathrm{Ref}(EF), \mathrm{SAT}^*)$ is $\leq_p$-hard for all DNPP
reductions	$(I_1(P), I_2(P)) \leq_p (U_1(P), U_2)$ $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (U_1(P), U_2)$ $\mathsf{DNPP}(P)$ is not closed under $\leq_p$ unless $(\mathrm{Ref}(EF), \mathrm{SAT}^*)$ is $\leq_p$-hard for all DNPP
properties	closed under modus ponens not closed under substitutions by constants unless $(\mathrm{Ref}(EF), \mathrm{SAT}^*)$ is $\leq_p$-hard for all DNPP

Table 4.1: The class $\mathsf{DNPP}(P)$ for different types of proof systems

NP-complete the NP-completeness of $\mathrm{Ref}(P)$ is a necessary condition for the opposite reduction to exist. To determine the complexity of $\mathrm{Ref}(P)$ for natural proof systems seems to be an interesting open problem. Approaching this question we note the following:

Proposition 4.15.1 *1. For every proof system P that is closed under disjunctions there is a proof system P' with $P' \equiv_p P$ such that $\mathrm{Ref}(P')$ is NP-complete.*

2. On the other hand there are proof systems P and P' such that $P \equiv_p P'$ and $\mathrm{Ref}(P)$ is decidable in polynomial time while $\mathrm{Ref}(P')$ is NP-complete.

Proof. To show part 1 of the proposition let P be a proof system that is closed under disjunctions. Closure under disjunctions implies in particular the existence of polynomial size proofs of all formulas of the form $\varphi \vee \top$ for arbitrary formulas φ. We define P' as

$$P'(\pi) = \begin{cases} P(\pi') & \text{if } \pi = 0^{q(|P(\pi')|)}1\pi' \\ \varphi \vee \top & \text{if } \pi = (\varphi, \alpha) \text{ and } \alpha \text{ is a satisfying assignment for } \varphi \\ \top & \text{otherwise} \end{cases}$$

with some polynomial q such that

$$q(n) \geq \max\{|(\varphi, \alpha)| \mid |\varphi \vee \top| = n\} \ .$$

Obviously P' is a correct proof system with $P \equiv_p P'$. Furthermore $\mathrm{Ref}(P')$ is NP-complete because SAT reduces to $\mathrm{Ref}(P')$ via

$$\varphi \mapsto (\varphi \vee \top, 1^{q(|\varphi \vee \top|)}) \ .$$

For part 2 we define the proof system P as follows: (π, φ) is a P-proof of φ, if either π is a correct truth-table evaluation of φ with all entries 1, or φ is of the form $\psi \vee \top$ for some formula ψ and $\pi = 1^{\|\mathrm{Var}(\psi)\|}$.

The proof system P satisfies the condition $P \vdash_* \psi \vee \top$ for all formulas ψ. Hence by the proof of part 1 of this proposition there is a proof system P' with $P \equiv_p P'$ and NP-complete $\mathrm{Ref}(P')$. On the other hand the set

$$\begin{aligned} \mathrm{Ref}(P) \ = \ & \{(\varphi, 1^m) \mid \varphi \in \mathrm{TAUT},\ m \geq 2^{\|\mathrm{Var}(\varphi)\|} + |\varphi|\} \cup \\ & \{(\psi \vee \top, 1^m) \mid \psi \text{ is a formula},\ m \geq \|\mathrm{Var}(\psi)\| + |\psi|\} \end{aligned}$$

is decidable in polynomial time. □

The second part of the above proposition tells us that the complexity of $\mathrm{Ref}(P)$ is not a robust property, i.e. it is not determined by the $\leq_p$-degree of the proof system P.

For strong systems P simulating bounded-depth Frege systems we know that the set $\mathrm{Ref}(P)$ cannot be decided in polynomial time unless for instance the RSA system is insecure (cf. Sect. 5.1). Hence the exact characterization of the complexity of $\mathrm{Ref}(P)$ seems to be an interesting open problem. Are those sets candidates for languages with complexity intermediate between P and NP-complete?

4.16 Are Canonical Pairs Something Special?

At this point it is the right time to discuss a recent result of Glaßer, Selman and Zhang (GSZ05). The last sections were devoted to a detailed analysis of canonical pairs, in particular of reductions between these pairs and their role for the subclasses $\mathsf{DNPP}(P)$. It is therefore natural to inquire whether canonical pairs enjoy special properties that distinguish them from other NP-pairs. The answer is given in a very general way by the following theorem:

Theorem 4.16.1 (Glaßer, Selman, Zhang (GSZ05))
Every disjoint NP*-pair is* $\leq_p$*-equivalent to the canonical pair of some propositional proof system.*

Before we describe the construction let us discuss two possible interpretations of this result. A first interpretation could be that canonical pairs do not seem to be anything special because every disjoint NP-pair essentially is a canonical pair. Therefore, for further investigation into NP-pairs we can dispense with the analysis of canonical pairs altogether and rather concentrate on the general concept of disjoint NP-pairs. Naturally, given the number of pages that we already devoted to canonical pairs in this dissertation this is not our favourite interpretation.

However, the result can also be understood as confirmation for the fact that propositional proof systems in the general definition of Cook and Reckhow (CR79) and disjoint NP-pairs are closely connected concepts from the same level of abstraction. So far, we have mostly used this connection to transfer information from proof systems to NP-pairs by associating various disjoint NP-pairs with a propositional proof system. The result of Glaßer et al. demonstrates that this transfer also works in the opposite direction in a very tight way: for every NP-pair there exists a proof system that captures the pair in the precise meaning of Theorem 4.16.1. The proof systems constructed for this purpose are just variants of the truth-table system. More precisely, for a given pair (A, B) a description of this pair is coded into the truth-table system. The drawback of this construction is that the proof

systems obtained in this way are rather artificial and in particular do not satisfy any of the natural closure properties that we have considered. However, proof systems that are used in practice and that are investigated in proof complexity usually satisfy these properties. Further, in the previous sections we illustrated that the canonical pairs of sufficiently well defined proof systems like regular proof systems are meaningful as complete pairs for some class of DNPP but that this property is lost for canonical pairs defined from arbitrary proof systems. These observations indicate that the Cook-Reckhow framework for propositional proof systems might be too broad for the study of naturally defined classes of disjoint NP-pairs (and in fact for other topics in proof complexity as well). It therefore seems to be natural to make additional assumptions on the properties of proof systems. Consequently, in our opinion, the canonical pairs of these natural proof systems deserve special attention.

We are now going to describe the construction of the proof system P from a given pair (A, B) as in the proof of Theorem 4.16.1. Because we are also interested in the stronger $\leq_s$-reduction we will analyse the construction under $\leq_s$. We note that we cannot expect a similar result as Theorem 4.16.1 for $\leq_s$ because $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_s (A, B)$ would imply a many-one-reduction from SAT^* to B and hence the NP-completeness of B which we did not assume. The opposite reduction, however, is $\leq_s$. Concerning the problem of the complexity of the set $\mathrm{Ref}(P)$ which we already addressed in Sect. 4.15 it is interesting to mention that the complexity of $\mathrm{Ref}(P)$ is determined by A, i.e. $\mathrm{Ref}(P)$ and A are many-one-equivalent. We combine this refined analysis in the next theorem. Its proof is essentially due to Glaßer, Selman and Zhang (GSZ05).

Theorem 4.16.2 *For every disjoint* NP*-pair* (A, B) *there exists a propositional proof system* P *such that the following holds:*

1. $(A, B) \leq_s (\mathrm{Ref}(P), \mathrm{SAT}^*)$.
2. $(\mathrm{Ref}(P), \mathrm{SAT}^*) \leq_p (A, B)$.
3. $A \equiv_m^p \mathrm{Ref}(P)$.

Proof. Let (A, B) be a disjoint NP-pair and let g be a polynomial time computable and polynomial time invertible many-one reduction from B to SAT. Let M be a nondeterministic Turing machine accepting A which runs in polynomial time specified by the polynomial p.

We define a proof system P as follows:

$$P(\langle x,w\rangle) = \begin{cases} \neg g(x) & |w| = p(|x|) \text{ and } M(x,w) \text{ accepts} \\ x & |w| \neq p(|x|), |w| \geq 2^{|x|}, x \in \text{TAUT} \\ \top & \text{otherwise} \end{cases}$$

Let us first argue that P is indeed a proof system. If w is of the correct length and $M(x,w)$ accepts, then $x \in A$, hence $x \notin B$ and therefore $g(x) \notin \text{SAT}$. Consequently $\neg g(x)$ is a tautology.

If $w \geq 2^{|x|}$, then we can check in polynomial time whether x is a tautology or not.

Hence P is computable in polynomial time and outputs only tautologies. But every tautology also has a P-proof of exponential size according to line 2 of the definition of P, so P is a proof system.

Let $q(|x|)$ be the precise length of $\langle x,w\rangle$ for inputs x,w satisfying $|w| = p(|x|)$. The function q is a polynomial for some suitable choice of the pairing function $\langle .,.\rangle$. We now claim that (A,B) is $\leq_s$-reducible to $(\text{Ref}(P), \text{SAT}^*)$ via the reduction

$$x \mapsto (\neg g(x), 1^{q(|x|)}) \ .$$

To see this let first $x \in A$. Then there exists a witness w of length $p(|x|)$ such that $M(x,w)$ accepts. Hence $(\neg g(x), 1^{q(|x|)}) \in \text{Ref}(P)$.

If $x \in B$, then $g(x)$ is satisfiable and therefore $(\neg g(x), 1^{q(|x|)}) \in \text{SAT}^*$.

Now let $x \notin A \cup B$. Since $x \notin A$ there is no witness w for x and $\neg g(x)$ cannot have a P-proof according to line 1 of the definition of P. A P-proof according to line 2 has length $2^{|x|} > q(|x|)$ and therefore $(\neg g(x), 1^{q(|x|)}) \notin \text{Ref}(P)$. From $x \notin B$ it follows that $g(x) \notin \text{SAT}$. Hence $(\neg g(x), 1^{q(|x|)}) \notin \text{SAT}^*$.

Thus we have proved

$$(A,B) \leq_s (\text{Ref}(P), \text{SAT}^*) \ .$$

Now we prove the reverse reduction

$$(\text{Ref}(P), \text{SAT}^*) \leq_p (A,B) \ .$$

Fix some elements $a \in A$ and $b \in B$. The reduction is performed by the following algorithm:

```
1  Input:  (x, 1^m)
2  IF x = ⊤ THEN output a
3  IF m ≥ 2^|x| THEN
4    IF x ∈ TAUT THEN output a ELSE output b
5  ELSE
6    IF g^-1(¬x) exists THEN output g^-1(¬x) ELSE output b
```

Let us argue that the reduction is correct. If x is a tautology different from $\top$, then there are two possibilities for proof lengths of x. Namely we have polynomial size proofs of size $q(|g^{-1}(\neg x)|)$ for formulas x where $g^{-1}(\neg x)$ exists and $g^{-1}(\neg x) \in A$, and proofs of exponential size $\geq 2^{|x|}$ for all other tautologies.

If $m \geq 2^{|x|}$, then the input $(x, 1^m)$ is correctly mapped according to line 4.

Consider now inputs $(x, 1^m)$ with $m < 2^{|x|}$. Let first $(x, 1^m) \in \mathrm{Ref}(P)$. Then x can only have a P-proof of size m if $g^{-1}(\neg x)$ exists in which case we output $g^{-1}(\neg x) \in A$ according to line 6. Tautologies which do not have this kind of proof are mapped to b. Therefore the reduction fails to be $\leq_s$.

Now suppose $(x, 1^m) \in \mathrm{SAT}^*$. Then $\neg x$ is satisfiable and the output is either $g^{-1}(\neg x) \in B$ or $b \in B$ according to line 6.

Finally, we will prove part 3 of the theorem. We have established already that (A, B) is $\leq_s$-reducible to $(\mathrm{Ref}(P), \mathrm{SAT}^*)$. Hence $A \leq_m^p \mathrm{Ref}(P)$ is given by the same reduction function $x \mapsto (\neg g(x), 1^{q(|x|)})$.

$\mathrm{Ref}(P)$ reduces to A by the following algorithm:

```
1  Input:  (x, 1^m)
2  IF x = ⊤ THEN output a
3  IF m ≥ 2^|x| THEN
4    IF x ∈ TAUT THEN output a ELSE output b
5  ELSE
6    IF g^{-1}(¬x) exists and m ≥ q(|g^{-1}(¬x)|) THEN
7      output g^{-1}(¬x)
8    ELSE output b
```

□

We already remarked that in essence the construction in the last proof codes the pair (A, B) into the truth-table system. Actually, we have frequently used a similar construction in previous sections. Namely, if (A, B) is a disjoint NP-pair and φ_n and ψ_n are propositional representations for A and B, respectively, then we can easily code (A, B) into a proof system P by augmenting P with polynomial size proofs of $\neg\varphi_n \vee \neg\psi_n$. For example, for the system EF this would result in

$$EF + \{\neg\varphi_n \vee \neg\psi_n \mid n \geq 0\} \ .$$

Clearly, we then have

$$(A, B) \leq_p (\mathrm{Ref}(EF + \{\neg\varphi_n \vee \neg\psi_n \mid n \geq 0\}), \mathrm{SAT}^*) \ .$$

However, whether the other reduction also holds is not clear, because the system $EF + \{\neg\varphi_n \vee \neg\psi_n \mid n \geq 0\}$ is a very strong system with good closure properties (cf. Proposition 2.6.6).

Chapter 5

Two Applications

Wissenschaften enfernen sich im Ganzen immer vom Leben und kehren nur durch einen Umweg wieder dahin zurück.

Johann Wolfgang Goethe

In this chapter we will describe two applications of the theory of disjoint NP-pairs. In the first application disjoint NP-pairs are used to model security aspects of crypto systems. As mentioned earlier this was the first motivation for the study of disjoint NP-pairs (ESY84; GS88; HS92).

The second application connects to a more recent line of research which aims to utilize pseudorandom generators for the construction of lower bounds to the lengths of proofs in strong propositional proof systems (Kra01b; Kra04; ABSRW04).

5.1 Security of Public-Key Crypto Systems

This section contains a brief description of some aspects of the relationship between public-key cryptosystems and disjoint NP-pairs. In fact, this connection was the starting point for the development of the theory of disjoint NP-pairs by Grollmann and Selman (GS88). We will not explain their results but only illustrate how disjoint NP-pairs can be defined from public-key cryptosystems.

One of the most common public-key cryptosystems is the RSA system developed by Rivest, Shamir and Adleman (RSA78). Let us briefly recall this cryptosystem. The public key consists of a number n which is the product of two primes together with an element e that is invertible modulo $\varphi(n)$. The private key is the inverse d of e modulo $\varphi(n)$. Encryption proceeds by raising

the plaintext x to the e-th power modulo n. Decryption of the ciphertext $y \equiv x^e \bmod n$ is accomplished by $x \equiv y^d \bmod n$.

Based on this cryptosystem Krajíček and Pudlák (KP98) defined a disjoint NP-pair (RSA_0, RSA_1) as follows:

$$\begin{aligned} RSA_0 &= \{(n,e,y,i) \mid \ (n,e) \text{ is a valid RSA key, } \exists x\, x^e \equiv y \bmod n \\ &\qquad\qquad \text{and the } i\text{-th bit of } x \text{ is } 0\} \\ RSA_1 &= \{(n,e,y,i) \mid \ \ldots \ \text{ is } 1\ \} \end{aligned}$$

By the phrase (n, e) is a valid RSA key we mean that n is the product of two primes p and q, and the public key e has a multiplicative inverse modulo $\varphi(n)$. By guessing the prime factorization $n = pq$ and determining $\varphi(n) = (p-1)(q-1)$ the validity of the public key (n, e) can be verified in nondeterministic polynomial time. Guessing further the plaintext x corresponding to the given ciphertext y and checking that x properly encrypts to y yields the value of the i-th bit of x. This shows that the components RSA_0 and RSA_1 are in NP. As their intersection is obviously empty we have defined a disjoint NP-pair.

The pair (RSA_0, RSA_1) has the additional property that also the complement of $RSA_0 \cup RSA_1$ is an NP-set. The complement contains all those inputs (n, e, y, i) where n, e does not form a valid RSA key. Again this can be verified by guessing the factorization of n and, in case n has exactly two prime factors, checking whether e is invertible modulo $\varphi(n)$.

Properties of this pair model the security of the RSA system. Namely, if the pair (RSA_0, RSA_1) is p-separable, then we can break the RSA by computing all ciphertext bits for a given plaintext. But also the converse is true, i.e. the pair (RSA_0, RSA_1) is p-separable if and only if we can compute to each ciphertext the corresponding plaintext in deterministic polynomial time without knowing the private key. But as already Grollmann and Selman discussed in their paper (GS88) worst-case complexity is not an appropriate measure for the security of cryptosystems. Namely, the p-inseparability of (RSA_0, RSA_1) might rest only on some hard instances while most ciphertexts are easy to decrypt. Therefore the p-separability of the RSA pair does not characterize the security of RSA. But of course the p-inseparability of (RSA_0, RSA_1) constitutes a necessary condition for the security of the RSA cryptosystem. This provides strong evidence that p-inseparable disjoint NP-pairs exist. Not only the RSA cryptosystem but in fact any one-way function gives rise to a disjoint NP-pair which is presumably not p-separable.

The link of such cryptographic pairs to propositional proof systems was established by Krajíček and Pudlák (KP98). In particular they demonstrated that the theory S_2^1 is sufficiently strong to prove the disjointness of the RSA

pair with respect to some natural representations of the components derived from the above definition of the pair.

Theorem 5.1.1 (Krajíček, Pudlák (KP98)) *The theory S_2^1 proves the disjointness of the pair (RSA_0, RSA_1).*

The proof which we skip involves verifying that the number-theoretic arguments used in the straightforward proof of the disjointness of the RSA pair formalize in S_2^1.

Using our terminology from the previous chapter we may rephrase this theorem as follows:

Corollary 5.1.2 *The pair (RSA_0, RSA_1) is representable in EF.*

In particular, this implies that the RSA-pair is $\leq_s$-reducible to the canonical pair of EF. Therefore, assuming the security of RSA, no proof system $P \geq EF$ can have a p-separable canonical pair. By Proposition 4.4.3 this also implies that none of these strong systems is automatizable.

As by Theorem 4.8.7 also the interpolation pair of EF is $\leq_s$-hard for $\mathsf{DNPP}(EF)$ we get the reduction

$$(RSA_0, RSA_1) \leq_s (I_1(EF), I_2(EF)) \ .$$

Therefore security of RSA implies that the interpolation pair of EF is not p-separable. By Theorem 4.4.8 this means that EF does not have feasible interpolation. In fact, this was the original motivation for Theorem 5.1.1. Subsequently it was shown that also Frege systems and bounded-depth Frege systems do not admit feasible interpolation under plausible assumptions (BPR00; BDG$^+$04).

5.2 Pseudorandom Generators in Proof Complexity

This section is devoted to a potential application of the results of the previous chapter for the construction of hard tautologies from pseudorandom generators (called *τ-formulas*). To employ pseudorandom generators as the basis for proving lower bounds to the proof size in propositional proof systems was independently suggested by Krajíček (Kra01a; Kra01b; Kra04) and by Alekhnovich, Ben-Sasson, Razborov and Wigderson (ABSRW04). These τ-formulas are candidates for tautologies without polynomially long proofs in strong proof systems like EF and their extensions. Proving super-polynomial lower bounds for strong proof systems constitutes a major open problem in

propositional proof complexity. The aim of this section is to illustrate that the hardness of τ-formulas can be expressed by properties of disjoint NP-sets.

We recall some terminology from (Kra04). Let $C = (C_n)_{n\in\mathcal{N}}$ be a family of polynomial size boolean circuits such that C_n is a circuit with n input and $m(n) > n$ output bits with some polynomial m. Functions f computed by such families C are called *polynomially stretching (p-stretching)*.

For $b \in \{0,1\}^{m(n)}$ we consider propositional formulas $\tau(C)_b$. The formula $\tau(C)_b$ has propositional variables $p_1, \ldots, p_n$ for the bits of the input of C_n, $q_1, \ldots, q_{m(n)}$ for the bits of the output of C_n and $r_1, \ldots, r_{n^{O(1)}}$ for the inner nodes of C_n. The formula $\tau(C)_b$ expresses that if $\bar{r}$ are correctly computed according to C_n from the input variables $\bar{p}$, then the values of the output variables $\bar{q}$ are different from the bits of b. The formula $\tau(C)_b$ is a tautology if and only if $b \notin \mathrm{rng}(f)$. But apparently $\tau(C)_b$ does not only depend on $\mathrm{rng}(f)$ but also on the particular circuits C_n used for the computation of f.

The formulas $\tau(C)$ from a circuit family C_n are called *hard* for a proof system P, if there does not exist a sequence of pairwise different numbers $b_n \in \{0,1\}^{m(n)}$, $n \in \mathcal{N}$, such that

$$P \vdash_* \tau(C)_{b_n} \ .$$

The intuition is that for functions having pseudorandom properties it should be hard to prove that a given element lies outside the range of the function. The hardness of a p-stretching function can be characterized by a hitting set property for NP/poly-sets. For this we need the following definition of the resultant of a p-stretching map.

Definition 5.2.1 (Krajíček (Kra04)) *Let f be a p-stretching map computed by the circuit family $C = (C_n)_{n\in\mathcal{N}}$ and let P be a propositional proof system. The* resultant of C with respect to P, *denoted by Res_C^P, consists of all* NP/poly*-sets A for which there exists a propositional representation $\varphi_n(\bar{x}, \bar{y})$ of A such that*

$$P \ \vdash_* \ \varphi_n(\bar{x}, \bar{y}) \to C(z) \neq x \ .$$

In (Kra04) this definition is formulated slightly differently, but as already here the close connection to disjoint NP-pairs becomes visible we have used similar terminology as in the previous chapters. The following theorem characterizes the hardness of τ-formulas by a condition on the resultant of P.

Theorem 5.2.2 (Krajíček (Kra04)) *Let P be a proof system of the form $EF + \Phi$ for some polynomial time computable set $\Phi \subseteq$ TAUT. Let f be a p-stretching function and C a polynomial size circuit family computing f. Then the following are equivalent:*

1. *The formulas $\tau(C)$ are hard for P.*
2. *The resultant Res_C^P contains only finite sets.*

Proof. For the first direction assume that the resultant contains an infinite NP/poly-set A that is represented by the propositional formulas $\varphi_n(\bar{x}, \bar{y})$. We choose a sequence of pairwise distinct elements a_i in A with $|a_i| = n_i$. By assumption we have

$$P \vdash_* \varphi_{n_i}(\bar{x}, \bar{y}) \to C(z) \neq x \ .$$

For $a_i \in A$ we now choose witnesses b_i with $|b_i| \leq |a_i|^k$ such that

$$\models \varphi_{n_i}(\bar{a}_i, \bar{b}_i) \ .$$

Because P is closed under substitutions by constants we obtain

$$P \vdash_* \varphi_{n_i}(\bar{a}_i, \bar{b}_i) \to C(z) \neq a_i \ .$$

Evaluating $\varphi_{n_i}(\bar{a}_i, \bar{b}_i)$ to $\top$ and applying modus ponens we arrive at

$$P \vdash_* C(z) \neq a_i \ .$$

Hence the formulas $\tau(C)_{a_i}$ have polynomial size proofs in P and therefore $\tau(C)$ is not hard for P.

For the opposite direction let us assume that the formulas $\tau(C)$ are not hard for P. Then there exists a polynomial p such that the NP/poly-set

$$A = \{a \in \{0,1\}^* \mid P \vdash_{\leq p(|a|)} \tau(C)_a\}$$

is infinite. As a propositional representation for A we can choose the formulas

$$\|\mathrm{Prf}_P(\pi, \tau(C)_a)\|^{p(|a|)} \ .$$

Using the reflection principle of P and modus ponens we obtain P-proofs of

$$\|\mathrm{Taut}(\tau(C)_a)\|^{|a|}$$

from which we conclude with Lemma 3.6.3 that $\tau(C)_a$ has polynomial size P-proofs for all $a \in A$. □

In fact the hardness of the function f should not depend on the particular circuits used for the computation of f. For functions f computed by non-uniform circuit families it is, however, not possible to get hard formulas $\tau(C)$ for all circuit families C computing f.

While this is not difficult to prove formally it is also intuitively clear. If a function f is computed by the circuits C which might yield hard formulas $\tau(C)$, then we can modify these circuits to a circuit family C' as follows. To the output gates of C we attach a circuit of polynomial size which compares the output produced by C with polynomially many fixed elements from the complement of $\mathrm{rng}(f)$. If this test is positive, then we output a fixed element from $\mathrm{rng}(f)$, otherwise we return the original output of C. Obviously, C and C' compute the same function f. But intuitively the formulas $\tau(C')$ are not hard for sufficiently strong proof systems P. By inspecting the extra gates attached to the circuits C we can devise short P-proofs for the disjointness of $\mathrm{rng}(f)$ and the set of those elements which are excluded in the extra gates of C'.

However, the situation is different for the functions $f \in \mathsf{FP}$ which are computed by uniform circuit families. Focusing therefore on the case where the circuit families are uniformly given we say that a polynomial time computable p-stretching function f yields *representationally independent hard τ-formulas* for P, if for every uniformly given circuit family C computing f the resulting formulas $\tau(C)$ are hard for P.

In this case also the resultant Res_C^P has to be defined efficiently and contains just NP-sets which are disjoint with $\mathrm{rng}(f)$ and where this disjointness is provable with short P-proofs. We can therefore use our terminology about disjoint NP-pairs to rephrase condition 2 of the theorem by the following condition 2':

2'. All sets $A \in \mathsf{NP}$ with $(A, \mathrm{rng}(C)) \in \mathsf{DNPP}(P)$ are finite.

We point out that in condition 2' the disjointness of A and $\mathrm{rng}(f)$ has to be proven with respect to the circuit family used for the computation of f, while the representation of A can be chosen arbitrarily.

Using the $\leq_s$-completeness of the U-pair for $\mathsf{DNPP}(P)$ (Theorem 4.8.6) we can restate Theorem 5.2.2 in the following form:

Corollary 5.2.3 *Let P be a proof system of the form $EF + \Phi$ for some polynomial time computable set $\Phi \subseteq \mathrm{TAUT}$. For every p-stretching function $f \in \mathsf{FP}$ the following are equivalent:*

1. *f yields representationally independent hard τ-formulas for P.*
2. *Every set $A \in \mathsf{NP}$ with $A \cap \mathrm{rng}(f) = \emptyset$ and $(A, \mathrm{rng}(f)) \leq_s (U_1(P), U_2)$ is finite.*

The difference between Corollary 5.2.3 and Theorem 5.2.2 is that condition 2 of the corollary only speaks about $\mathrm{rng}(f)$ whereas condition 2 of the above theorem involves the particular circuits used for the computation of f.

Dropping the condition $(A, \mathrm{rng}(f)) \leq_s (U_1(P), U_2)$ from condition 2 of the corollary we arrive at an NP-set $B = \mathrm{rng}(f)$ containing no infinite NP-set in its complement $\bar{B}$. Such sets B are called NP-simple (see (BDG88) or (SY04)). By Corollary 5.2.3 NP-simple sets would yield representationally independent hard τ-formulas for all proof systems, but their existence is open.

Simplicity is a concept originating in recursion theory that can be defined for any complexity class.

Definition 5.2.4 *Let* C *be a complexity class.*

1. *A set A is called* C-immune *if every subset $B \subseteq A$ with $B \in \mathsf{C}$ is finite.*
2. *A is called* C-simple*, if $A \in \mathsf{C}$ and $\bar{A}$ is* C*-immune.*

Here we are interested in the cases $\mathsf{C} = \mathsf{P}$ and $\mathsf{C} = \mathsf{NP}$. As mentioned the question whether NP-simple sets exist is open. Obviously $\mathsf{NP} \neq \mathsf{coNP}$ is a necessary condition for the existence of NP-simple sets, other necessary or sufficient conditions are, however, not known. Vereshchagin proved that NP-simple sets exist relative to a random oracle (Ver95).

What we actually need for the hardness of τ-formulas is not the existence of NP-simple sets, but a weaker condition which could be formalized as:

Definition 5.2.5 *Let (C, D) be a disjoint* NP*-pair. We call a set A* NP-simple relative to (C, D) *if $A \in \mathsf{NP}$ and for all infinite sets $B \in \mathsf{NP}$ with $A \cap B = \emptyset$ we have $(A, B) \not\leq_s (C, D)$.*

With this definition Corollary 5.2.3 takes the following form:

Corollary 5.2.6 *For all proof systems $P = EF + \Phi$ with polynomial time computable $\Phi \subseteq$ TAUT and all p-stretching functions $f \in \mathsf{FP}$ the following are equivalent:*

1. *f yields representationally independent hard τ-formulas for P.*
2. *$\mathrm{rng}(f)$ is* NP*-simple relative to $(U_1(P), U_2)$.*

The following easy proposition gives a characterization of the relative simplicity of an NP-set.

Proposition 5.2.7 *Let $A \in \mathsf{NP}$ and let (C, D) be a disjoint* NP*-pair. Then A is* NP*-simple relative to (C, D) if and only if for all $\leq^p_m$-reductions g : $A \leq^p_m C$ the set $g^{-1}(D)$ is finite.*

Proof. Let A be NP-simple relative to (C, D). Let us assume that $g^{-1}(D)$ is infinite for some reduction $g : A \leq_m^p C$. We have $g^{-1}(D) \in \mathsf{NP}$ and $A \cap g^{-1}(D) = \emptyset$. Therefore g reduces the disjoint NP-pair $(A, g^{-1}(D))$ to (C, D), i.e. A is not NP-simple relative to (C, D).

If on the contrary A is not NP-simple relative to (C, D), then there exists an infinite set $B \in \mathsf{NP}$ with $A \cap B = \emptyset$ and $g : (A, B) \leq_s (C, D)$ via some function $g \in \mathsf{FP}$. Then $g^{-1}(D)$ contains B and is therefore infinite. □

The proof of Proposition 5.2.7 also makes it clear that the relative NP-simplicity of a set does not depend on the strength of the reduction used, i.e. using the weaker reduction $\leq_p$ instead of $\leq_s$ in Definition 5.2.5 results in the same concept.

In view of the above proposition the NP-simplicity of A relative to (C, D) can also come from the fact that A is not $\leq_m^p$-reducible to C. But for the case where $(C, D) = (U_1(P), U_2)$ this cannot happen as $U_1(P)$ and U_2 are NP-complete. In this case we can give the following necessary condition for the relative NP-simplicity of A.

Proposition 5.2.8 *Let A be NP-simple relative to (C, D) and let A be $\leq_m^p$-reducible to C. Then $\bar{A}$ is P-immune.*

Proof. Let $g : A \leq_m^p C$. If $\bar{A}$ is not P-immune, then there exists an infinite set $B \in \mathsf{P}$ with $A \cap B = \emptyset$. Then the disjoint NP-pair (A, B) is $\leq_s$-reducible to (C, D) via

$$g'(x) = \begin{cases} g(x) & \text{if } x \notin B \\ x_0 \in D & \text{if } x \in B, \end{cases}$$

i.e. A is not NP-simple relative to (C, D). □

Therefore the relative NP-simplicity of a set A is a notion which lies in strength between the P-immunity of the complement $\bar{A}$ and the NP-simplicity of A. Whether disjoint NP-pairs will indeed prove to be helpful in establishing lower bounds to the proof size in strong proof systems must remain open. The characterization of these difficult proof-theoretic problems in terms of disjoint NP-pair as given in Corollary 5.2.3 shows, however, that investigation into the structure of NP-pairs will remain a demanding and potentially rewarding task.

Chapter 6

Disjoint Tuples of NP-Sets

Aus vielen Skizzen endlich ein Ganzes hervorbringen gelingt selbst den Besten nicht immer.

Johann Wolfgang Goethe

In the previous chapters we have seen that disjoint NP-pairs are a natural concept with meaningful applications to cryptography and the theory of propositional proof systems. At this point it is a natural question for the enquiring mathematical mind to ask: can we generalize this to k-tuples and develop a corresponding theory of disjoint k-tuples of NP-sets? But also in many applications we find situations where not only two but a greater number of different, mutually exclusive conditions is of interest.

Hence this chapter is devoted to a generalization of the results from Chap. 4 to disjoint k-tuples of NP-sets. As many definitions and results are generalized in a straightforward manner we will explain the material in a more condensed form.

6.1 Basic Definitions and Properties

Definition 6.1.1 *Let $k \geq 2$ be a natural number. A tupel $(A_1, \ldots, A_k)$ is a disjoint k-tuple of* NP*-sets if all components $A_1, \ldots, A_k$ are nonempty languages in* NP *which are pairwise disjoint.*

We generalize the notion of a separator of a disjoint NP-pair in the following way:

Definition 6.1.2 *A function $f : \{0,1\}^* \to \{1, \ldots, k\}$ is a* separator *for a disjoint k-tuple $(A_1, \ldots, A_k)$ of* NP*-sets if for all $a \in \{0,1\}^*$*

$$a \in A_i \quad \Longrightarrow \quad f(a) = i \quad \text{for } i = 1, \ldots, k \ .$$

For inputs from the complement $\overline{A_1 \cup \ldots \cup A_k}$ *the function* f *may answer arbitrarily.*

If $(A_1, \ldots, A_k)$ *is a disjoint* k*-tuple of* NP*-sets that has a polynomial time computable separator we call the tuple* p-separable*, otherwise* p-inseparable*.*

Whether there exist p-inseparable disjoint k-tuples of NP-sets is a certainly a hard problem that cannot be answered with our current techniques. At least we can show that this question is not harder than the previously studied question whether there exist p-inseparable disjoint NP-pairs.

Theorem 6.1.3 *The following are equivalent:*

1. *For all natural numbers* $k \geq 2$ *there exist p-inseparable disjoint* k*-tuples of* NP*-sets.*

2. *There exists a natural number* $k \geq 2$ *such that there exist p-inseparable disjoint* k*-tuples of* NP*-sets.*

3. *There exist p-inseparable disjoint* NP*-pairs.*

Proof. Trivially, 1 implies 2. We will show $2 \Rightarrow 3$ and $3 \Rightarrow 1$.

In order to prove $2 \Rightarrow 3$ let us assume that all disjoint NP-pairs are p-separable. Let $k \geq 2$ be some number and $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets. By assumption we have separators $f_{i,j}$ for all disjoint NP-pairs (A_i, A_j) with $i, j \in \{1, \ldots, k\}$, $i \neq j$. We devise a separator for $(A_1, \ldots, A_k)$ as follows: at input a we first evaluate all functions $f_{i,j}(a)$. If there exists a number i such that we received 1 at all evaluations $f_{i,j}(a)$ for $j \in \{1, \ldots, k\} \setminus \{i\}$, then we output this number i. If no such i exists, then we know that a is outside $A_1 \cup \ldots \cup A_k$, and we can answer arbitrarily. If on the other hand $a \in A_i$, then we always get $f_{i,j}(a) = 1$ for $j \in \{1, \ldots, k\} \setminus \{i\}$. As only one such i can exist we produce the correct answer.

To show the remaining implication $3 \Rightarrow 1$ let us assume that the disjoint NP-pair (A, B) is p-inseparable. Without loss of generality we may assume that $\overline{A \cup B}$ is infinite because otherwise the pair (A, B) can be trivially modified to a p-inseparable pair that meets this condition. For a given number k let $a_3, \ldots, a_k$ be distinct elements from $\overline{A \cup B}$. Then $(A, B, \{a_3\}, \ldots, \{a_k\})$ is a p-inseparable disjoint k-tuple of NP-sets. □

Let us pause to give an example of a disjoint k-tuple of NP-sets that is derived from the Clique-Colouring pair. The tuple $(C_1, \ldots, C_k)$ has components of the following form:

$$C_i = \{G \mid G \text{ is an } i+1\text{-colourable graph with a clique of size } i\} \ .$$

Clearly, the components C_i are NP-sets which are pairwise disjoint. The tuple $(C_1, \ldots, C_k)$ is also p-separable, but to devise a separator for $(C_1, \ldots, C_k)$ is considerably simpler than to separate the Clique-Colouring pair: given a graph G we output the maximal number i between 1 and k such that G contains a clique of size i. For graphs with n vertices this number i can be computed in time $O(n^k)$. It would be nicer to define the components C_i by the requirement that the chromatic number of the graph G should be exactly $i+1$. This, however, would increase the complexity of C_i to $\mathsf{NP} \cup \mathsf{coNP}$. The situation is similar for asking for the exact value of other graph parameters that are not easily computable in polynomial time.

Candidates for p-inseparable tuples arise from one-way functions. Let $\Sigma = \{a_1, \ldots, a_k\}$ be an alphabet of size $k \geq 2$. To a one-way function $f : \Sigma^* \to \Sigma^*$ we assign a disjoint k-tuple $(A_1(f), \ldots, A_k(f))$ of NP-sets with components

$$A_i(f) = \{(y, j) \mid (\exists x) f(x) = y \text{ and } x_j = a_i\}$$

where x_j is the j-th letter of x. This tuple is p-inseparable if f has indeed the one-way property.

Next we define reductions for k-tuples. We will only consider variants of many-one reductions which are easily obtained from the reductions $\leq_p$ and $\leq_s$ for pairs. As there is no danger of confusion we will use the same symbols $\leq_p$ and $\leq_s$ for the generalized versions.

Definition 6.1.4 *Let $(A_1, \ldots, A_k)$ and $(B_1, \ldots, B_k)$ be disjoint k-tuples of* NP*-sets. We say that $(A_1, \ldots, A_k)$ is* polynomially reducible *to $(B_1, \ldots, B_k)$, denoted by*

$$(A_1, \ldots, A_k) \leq_p (B_1, \ldots, B_k) \ ,$$

if there exists a polynomial time computable function f such that $f(A_i) \subseteq B_i$ for all $i = 1, \ldots, k$.

The tuple $(A_1, \ldots, A_k)$ is strongly reducible *to $(B_1, \ldots, B_k)$, denoted by*

$$(A_1, \ldots, A_k) \leq_s (B_1, \ldots, B_k) \ ,$$

if there exists a polynomial time computable function f such that f performs a $\leq_p$-reduction from $(A_1, \ldots, A_k)$ to $(B_1, \ldots, B_k)$ and additionally $f(\overline{A_1 \cup \ldots \cup A_k}) \subseteq \overline{B_1 \cup \ldots \cup B_k}$.

As before we define from $\leq_p$ and $\leq_s$ equivalence relations $\equiv_p$ and $\equiv_s$ and call their equivalence classes degrees.

We call a disjoint k-tuple of NP-sets $\leq_p$*-complete* or $\leq_s$*-complete* if all disjoint k-tuples of NP-sets are $\leq_p$- or $\leq_s$-reducible to it.

As for pairs we observe that the complexity of the components of a k-tuple inside a $\leq_p$-degree can change while this is not possible for $\leq_s$-degrees.

Proposition 6.1.5 *1. For every disjoint k-tuple $(A_1, \ldots, A_k)$ of NP-sets there exists a disjoint k-tuple $(B_1, \ldots, B_k)$ of NP-sets such that*

$$(A_1, \ldots, A_k) \equiv_p (B_1, \ldots, B_k)$$

and $B_1, \ldots, B_k$ are NP-complete.

2. If f is a $\leq_s$-reduction between the disjoint k-tuples $(A_1, \ldots, A_k)$ and $(B_1, \ldots, B_k)$, then f is a many-one reduction from A_i to B_i for every $i = 1, \ldots, k$.

Proof. For part 1 choose $B_i = A_i \times \mathrm{SAT}$. Part 2 follows immediately from the definition of $\leq_s$. □

The difference between $\leq_p$ and $\leq_s$ as expressed in Proposition 6.1.5 allows us to separate the reductions $\leq_p$ and $\leq_s$ on the domain of all p-separable disjoint k-tuples of NP-sets:

Theorem 6.1.6 *For all natural numbers $k \geq 2$ the following holds:*

1. *All p-separable disjoint k-tuples of NP-sets are $\leq_p$-equivalent.*
2. *If $\mathsf{P} \neq \mathsf{NP}$, then there exist infinitely many $\leq_s$-degrees of p-separable disjoint k-tuples of NP-sets.*
3. *$\mathsf{P} \neq \mathsf{NP}$ if and only if there exist disjoint k-tuples $(A_1, \ldots, A_k)$ and $(B_1, \ldots, B_k)$ such that $\overline{A_1 \cup \ldots \cup A_k}$ and $\overline{B_1 \cup \ldots \cup B_k}$ are nonempty and $(A_1, \ldots, A_k) \leq_p (B_1, \ldots, B_k)$, but $(A_1, \ldots, A_k) \not\leq_s (B_1, \ldots, B_k)$.*

Proof. Parts 1 and 2 are proved analogously as Propositions 4.2.1 and 4.2.4. Part 3 is a consequence of parts 1 and 2. □

6.2 Representable Disjoint Tuples of NP-Sets

Definition 6.2.1 *Let P be a propositional proof system. A disjoint k-tuple $(A_1, \ldots, A_k)$ of NP-sets is* representable in P *if there exist propositional representations $\varphi_n^i(\bar{x}, \bar{y}^i)$ of A_i for $i = 1, \ldots, k$ such that for each $1 \leq i < j \leq k$*

the formulas $\varphi_n^i(\bar{x}, \bar{y}^i)$ *and* $\varphi_n^j(\bar{x}, \bar{y}^j)$ *have only the variables* $\bar{x}$ *in common, and further*

$$P \vdash_* \bigwedge_{1 \le i < j \le k} \neg\varphi_n^i(\bar{x}, \bar{y}^i) \vee \neg\varphi_n^j(\bar{x}, \bar{y}^j) \ .$$

By $\mathsf{DNPP}_k(P)$ *we denote the class of all disjoint* k*-tuples of* NP*-sets which are representable in* P*.*

Because the classes $\mathsf{DNPP}_k(P)$ provide natural generalizations of the classes $\mathsf{DNPP}(P)$ we have chosen the same notation for the classes of k-tuples.

As in Sect. 4.5 we can show that the class $\mathsf{DNPP}_k(P)$ is closed under reductions.

Proposition 6.2.2 *Let* P *be a proof system that is closed under conjunctions and disjunctions and that simulates resolution. Then for all numbers* $k \ge 2$ *the class* $\mathsf{DNPP}_k(P)$ *is closed under* $\le_p$*.*

Proof. Let $(A_1, \ldots, A_k)$ and $(B_1, \ldots, B_k)$ be disjoint k-tuples of NP-sets such that f is a $\le_p$-reduction from $(A_1, \ldots, A_k)$ to $(B_1, \ldots, B_k)$. Let further P be a propositional proof system satisfying the above conditions and let $(B_1, \ldots, B_k) \in \mathsf{DNPP}_k(P)$.

Closure of P under conjunctions implies that for all $1 \le i < j \le k$ each of the disjoint NP-pairs (B_i, B_j) is contained in $\mathsf{DNPP}(P)$. As f is also a $\le_p$-reduction between the disjoint NP-pairs (A_i, A_j) and (B_i, B_j) we infer with Proposition 4.6.1 that all pairs (A_i, A_j) are in $\mathsf{DNPP}(P)$. Going back to the proof of Proposition 4.6.1 we see that P proves the disjointness of these pairs with respect to the representations

$$A_i' = \{x \mid x \in A_i \text{ and } f(x) \in B_i\} \ .$$

In particular, the representation of A_i is always the same when proving the disjointness of A_i and A_j for different j. Therefore we can combine these proofs of disjointness by conjunctions and obtain a P-proof of a suitable propositional description of

$$\bigwedge_{1 \le i < j \le k} A_i' \cap A_j' = \emptyset \ .$$

This shows $(A_1, \ldots, A_k) \in \mathsf{DNPP}_k(P)$. □

6.3 Disjoint Tuples of NP-Sets from Propositional Proof Systems

In this section we want to associate tuples of NP-sets with proof systems. It is not clear how the canonical pair could be modified for k-tuples but the interpolation pair as well as the U-pair can be stretched to more than two components. We start with the generalization of the U-pair.

For a propositional proof system P we define a k-tuple $(U_1(P), \ldots, U_k(P))$ with the components

$$U_i(P) = \{(\varphi_1, \ldots, \varphi_k, 1^m) \mid \operatorname{Var}(\varphi_j) \cap \operatorname{Var}(\varphi_l) = \emptyset \text{ for all } 1 \leq j < l \leq k, \\ \neg\varphi_i \in \mathrm{SAT} \text{ and } P \vdash_{\leq m} \bigwedge_{1 \leq j < l \leq k} \varphi_j \vee \varphi_l\}$$

for $i = 1, \ldots, k$. It is clear that all components $U_i(P)$ are in NP. To see their pairwise disjointness assume that $(\varphi_1, \ldots, \varphi_k, 1^m) \in U_i(P)$ and let $j \in \{1, \ldots, k\} \setminus \{i\}$. Because we have a P-proof of

$$\bigwedge_{1 \leq j < l \leq k} \varphi_j \vee \varphi_l \ ,$$

this formula is a tautology. Therefore in particular $\varphi_i \vee \varphi_j$ is a tautology and because φ_i and φ_j have no common variables either of these formulas must be tautological. As in the definition of $U_i(P)$ this is excluded for φ_i the formula φ_j is a tautology. But this implies $(\varphi_1, \ldots, \varphi_k, 1^m) \notin U_j(P)$.

Similarly, we can expand the interpolation pair of proof system to a k-tuple $(I_1(P), \ldots, I_k(P))$ by setting

$$I_i(P) = \{(\varphi_1, \ldots, \varphi_k, \pi) \mid \operatorname{Var}(\varphi_j) \cap \operatorname{Var}(\varphi_l) = \emptyset \text{ for all } 1 \leq j < l \leq k, \\ \neg\varphi_i \in \mathrm{SAT} \text{ and } P(\pi) = \bigwedge_{1 \leq j < l \leq k} \varphi_j \vee \varphi_l\}$$

for $i = 1, \ldots, k$. The same argument as above shows that $(I_1(P), \ldots, I_k(P))$ is indeed a disjoint k-tuple of NP-sets. Further, this tuple still captures the feasible interpolation property of the proof system P as the next theorem shows.

Theorem 6.3.1 *Let P be a propositional proof system that is efficiently closed under substitutions by constants and conjunctions. Likewise suppose we can efficiently modify a P-proof of an implication $\varphi \to \psi$ to a P-proof of $\neg\varphi \vee \psi$ and vice versa.*

Then $(I_1(P), \ldots, I_k(P))$ is p-separable if and only if P has the feasible interpolation property.

Proof. Because we assumed that P is efficiently closed under substitutions by constants and can handle implications we know by Theorem 4.4.8 that feasible interpolation of P is equivalent to the p-separability of $(I_1(P), I_2(P))$. It is therefore sufficient to show that for every $k \geq 2$ the pair $(I_1(P), I_2(P))$ is p-separable if and only if $(I_1(P), \ldots, I_k(P))$ is p-separable.

For the first direction assume that $(I_1(P), I_2(P))$ is separated by the polynomial time computable function f, i.e.

$$\begin{aligned} (\varphi, \psi, \pi) \in I_1(P) \quad &\Longrightarrow \quad f(\varphi, \psi, \pi) = 1 \\ (\varphi, \psi, \pi) \in I_2(P) \quad &\Longrightarrow \quad f(\varphi, \psi, \pi) = 0 \ . \end{aligned}$$

We separate the tuple $(I_1(P), \ldots, I_k(P))$ by the following algorithm: at input $(\varphi_1, \ldots, \varphi_k, \pi)$ we test whether π is indeed a P-proof of

$$\bigwedge_{1 \leq i < j \leq k} \varphi_i \vee \varphi_j \ .$$

If this is the case we can use the assumption that P is efficiently closed under conjunctions to compute P-proofs $\pi_{i,j}$ of $\varphi_i \vee \varphi_j$ for all $i, j \in \{1, \ldots, k\}$, $i \neq j$. We then test whether there exists an $i \in \{1, \ldots, k\}$ such that for all $j \in \{1, \ldots, k\} \setminus \{i\}$ we have $f(\varphi_i, \varphi_j, \pi_{i,j}) = 1$. If such i exists, then we output this number i.

It is clear that this algorithm runs in polynomial time. To see the correctness of the algorithm assume that $(\varphi_1, \ldots, \varphi_k, \pi) \in I_i(P)$. Then $\neg\varphi_i$ is satisfiable and hence $\varphi_1, \ldots, \varphi_{i-1}, \varphi_{i+1}, \ldots, \varphi_k$ are tautologies. Therefore $f(\varphi_i, \varphi_j, \pi_{i,j})$ always outputs 1. As this can happen for at most one i we give the correct answer.

For the converse direction assume that $(I_1(P), \ldots, I_k(P))$ is separated by the polynomial time computable function f, i.e.

$$(\varphi_1, \ldots, \varphi_k, \pi) \in I_i(P) \quad \Longrightarrow \quad f(\varphi, \ldots, \varphi_k, \pi) = i$$

for $i = 1, \ldots, k$. Let (φ, ψ, π) be given. We first check whether $P(\pi) = \varphi \vee \psi$. If this is fulfilled we expand (φ, ψ) to the k-tuple

$$(\varphi_1, \ldots, \varphi_k) = (\varphi, \psi, \top, \ldots, \top) \ .$$

We then use the assumption that P is efficiently closed under conjunctions to generate a P-proof π' of $\bigwedge_{1 \leq i < j \leq k} \varphi_i \vee \varphi_j$ from π. Finally, we evaluate $f(\varphi, \psi, \top, \ldots, \top, \pi')$. We use this answer to decide (φ, ψ, π), i.e. on output 1 we also answer with 1 and on output 2 we answer with 0. □

The next theorem is a generalization of Theorem 4.8.3 to k-tuples.

Theorem 6.3.2 *Let P be a proof system that is closed under substitutions by constants. Then for every $k \geq 2$ the k-tuple $(U_1(P), \ldots, U_k(P))$ is $\leq_s$-hard for $\mathsf{DNPP}_k(P)$.*

Proof. Let $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets and let $\varphi_n^i(\bar{x}, \bar{y}^i)$ be propositional representations of A_i for $i = 1, \ldots, k$ such that

$$P \vdash_* \bigwedge_{1 \leq i < j \leq k} \neg\varphi_n^i(\bar{x}, \bar{y}^i) \vee \neg\varphi_n^j(\bar{x}, \bar{y}^j) \ .$$

We claim that there exists a polynomial p such that

$$a \mapsto (\neg\varphi_{|a|}^1(\bar{a}, \bar{y}^1), \ldots, \neg\varphi_{|a|}^k(\bar{a}, \bar{y}^k), 1^{p(|a|)})$$

realizes a $\leq_s$-reduction from $(A_1, \ldots, A_k)$ to $(U_1(P), \ldots, U_k(P))$.

Verifying this claim proceeds similarly as in the proof of Theorem 4.8.3. □

For technical reasons we now introduce a modification $(V_1(P), \ldots, V_k(P))$ of the U-tuple for which we will also show the hardness for $\mathsf{DNPP}_k(P)$. Instead of k-tuples the components $V_r(P)$ now consist of sequences of $(k-1)k$ formulas together with an unary coded parameter m. For a propositional proof system P we define the k-tuple $(V_1(P), \ldots, V_k(P))$ as:

$$\begin{aligned} V_r(P) \ = \ & \{((\varphi_{i,j} \mid 1 \leq i, j \leq k, i \neq j), 1^m) \mid \\ & \quad \mathrm{Var}(\varphi_{i,j}) \cap \mathrm{Var}(\varphi_{l,n}) = \emptyset \text{ for all } i, j, l, n \in \{1, \ldots, k\}, i \neq l, \\ & \quad \neg\varphi_{r,i} \in \mathrm{SAT} \text{ for } i \in \{1, \ldots, k\} \setminus \{r\} \text{ and} \\ & \quad P \vdash_{\leq m} \bigwedge_{i=1}^{k} \bigwedge_{j=i+1}^{k} \varphi_{i,j} \vee \varphi_{j,i}\} \end{aligned}$$

for $r = 1, \ldots, k$. Let us verify that we have defined a disjoint k-tuple of NP-sets. It is clear that all components $V_r(P)$ are in NP. To prove their disjointness assume that the tuple $((\varphi_{i,j} \mid 1 \leq i, j \leq k, i \neq j), 1^m)$ is contained both in $V_r(P)$ and $V_s(P)$ for $r, s \in \{1, \ldots, k\}$, $r < s$. The definition of V_r guarantees that

$$\bigwedge_{i=1}^{k} \bigwedge_{j=i+1}^{k} \varphi_{i,j} \vee \varphi_{j,i}$$

is a tautology. Therefore in particular $\varphi_{r,s} \vee \varphi_{s,r}$ is a tautology and because $\varphi_{r,s}$ and $\varphi_{s,r}$ have no common variables either of these formulas must be tautological. In the definition of $V_r(P)$ this is excluded for $\varphi_{r,s}$ and in the definition of $V_s(P)$ this is excluded for $\varphi_{s,r}$ which gives a contradiction.

As this V-tuple is a generalization of the previously defined U-tuple we can reduce the U-tuple to the V-tuple, thereby showing the hardness result for the V-tuple:

Proposition 6.3.3 *Let P be a proof system that is closed under substitutions by constants. Then for every $k \geq 2$ the pair $(V_1(P), \ldots, V_k(P))$ is $\leq_s$-hard for $\mathsf{DNPP}_k(P)$.*

Proof. By Theorem 6.3.2 we know that $(U_1(P), \ldots, U_k(P))$ is $\leq_s$-hard for $\mathsf{DNPP}_k(P)$ for proof systems P that are closed under substitutions by constants. Therefore, to prove the result it is sufficient to $\leq_s$-reduce the U-tuple to $(V_1(P), \ldots, V_k(P))$. The reduction is given by

$$f : (\varphi_1, \ldots, \varphi_k, 1^m) \mapsto (\underbrace{\varphi_1, \ldots, \varphi_1}_{k-1}, \underbrace{\varphi_2, \ldots, \varphi_2}_{k-1}, \ldots, \underbrace{\varphi_k, \ldots, \varphi_k}_{k-1}, 1^m) \ .$$

To prove the correctness of the reduction it is enough to observe that for each $i = 1 \ldots, k$ we have $(\varphi_1, \ldots, \varphi_k, 1^m) \in U_i(P)$ if and only if $f(\varphi_1, \ldots, \varphi_k, 1^m) \in V_i(P)$. This is true because the conditions on the satisfiability and the disjointness of the variables of the formulas are trivially preserved, and the formulas

$$\bigwedge_{1 \leq j < l \leq k} \varphi_j \vee \varphi_l = \bigwedge_{j=1}^{k} \bigwedge_{l=j+1}^{k} \varphi_j \vee \varphi_l$$

which should be P-provable in size $\leq m$ are equal. □

6.4 Arithmetic Representations

As for disjoint NP-pairs we can also generalize the notion of arithmetic representations to disjoint k-tuples of NP-sets.

Definition 6.4.1 *A disjoint k-tuple $(A_1, \ldots, A_k)$ of NP-sets is* representable *in an L-theory T if there are Σ^b_1-formulas $\varphi_1(x), \ldots, \varphi_k(x)$ representing the components $A_1, \ldots, A_k$ such that*

$$T \vdash (\forall x) \bigwedge_{1 \leq i < j \leq k} \neg\varphi_i(x) \vee \neg\varphi_j(x) \ .$$

By $\mathsf{DNPP}_k(T)$ we denote the class of all disjoint k-tuples of NP-sets that are representable in T.

Similarly as in Theorem 4.5.8 we can show that also for k-tuples these uniformly defined classes coincide with the non-uniformly defined classes $\mathsf{DNPP}_k(P)$ for regular proof systems P corresponding to the theory T.

Theorem 6.4.2 *Let $P \geq EF$ be a regular proof system which is closed under substitutions by constants and conjunctions and let $T \supseteq S^1_2$ be a theory corresponding to T. Then we have $\mathsf{DNPP}_k(P) = \mathsf{DNPP}_k(T)$ for all $k \geq 2$.*

Proof. To show $\mathsf{DNPP}_k(P) \subseteq \mathsf{DNPP}_k(T)$ let $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets in $\mathsf{DNPP}_k(P)$ and let φ^i_n be propositional representations of the sets A_i for $i = 1, \ldots, k$, such that

$$P \vdash_* \bigwedge_{1 \leq i < j \leq k} \neg\varphi^i_n \vee \neg\varphi^j_n \ . \tag{6.1}$$

Because P is closed under conjunctions this in particular means

$$P \vdash_* \neg\varphi^i_n \vee \neg\varphi^j_n$$

for all $1 \leq i < j \leq k$, i.e. all disjoint NP-pairs (A_i, A_j) are contained in $\mathsf{DNPP}(P)$. By Proposition 4.5.7 this implies that for all $1 \leq i < j \leq k$ we have $(A_i, A_j) \in \mathsf{DNPP}(T)$ where the disjointness of (A_i, A_j) is T-provable via arithmetic representations $\psi_i(x)$ for A_i depending only on the set A_i and the polynomial in (6.1). Hence we get

$$T \vdash (\forall x) \bigwedge_{1 \leq i < j \leq k} \neg\psi_i(x) \vee \neg\psi_j(x) \tag{6.2}$$

and therefore $(A_1, \ldots, A_k) \in \mathsf{DNPP}_k(T)$

For the other inclusion let $\psi_1(x), \ldots, \psi_k(x)$ be arithmetic representations of $A_1, \ldots, A_k$ such that (6.2) holds. Then the translations $\|\psi_i(x)\|^n$ of the arithmetic representations ψ_i provide propositional representations of A_i for $i = 1, \ldots, k$. In these translations we choose the auxiliary variables disjoint. Because $\bigwedge_{1 \leq i < j \leq k} \neg\psi_i(x) \vee \neg\psi_j(x)$ is a Π^b_1-formula we get from (6.2)

$$P \vdash_* \| \bigwedge_{1 \leq i < j \leq k} \neg\psi_i(x) \vee \neg\psi_j(x)\|^n \ .$$

By definition of the translation $\|.\|$ this is equivalent to

$$P \vdash_* \bigwedge_{1 \leq i < j \leq k} \neg\|\psi_i(x)\|^n \vee \neg\|\psi_j(x)\|^n$$

and therefore $(A_1, \ldots, A_k) \in \mathsf{DNPP}_k(P)$. □

As for the case $k = 2$ we now observe that the k-tuples $(U_1(P), \ldots, U_k(P))$ and $(I_1(P), \ldots, I_k(P))$ are representable in P.

Lemma 6.4.3 *Let P be a regular proof system. Then for all numbers $k \geq 2$ the k-tuples $(U_1(P), \ldots, U_k(P))$, $(V_1(P), \ldots, V_k(P))$ and $(I_1(P), \ldots, I_k(P))$ are representable in P.*

Proof. Let P be regular and T be a theory associated with P. We show the representability of the tuples $(U_1(P), \ldots, U_k(P))$, $(V_1(P), \ldots, V_k(P))$ and $(I_1(P), \ldots, I_k(P))$ in T.

As arithmetic representations for the components $U_i(P)$, $V_i(P)$ and $I_i(P)$ we choose straightforward first-order formalizations which use the formulas Taut and Prf_P. Using the reflection principle of P which is available in T we can devise T-proofs of the arithmetic formalizations of $U_i(P) \cap U_j(P) = \emptyset$, $V_i(P) \cap V_j(P) = \emptyset$ and $I_i(P) \cap I_j(P) = \emptyset$ for all $1 \leq i < j \leq k$. Combining these proofs we get the representability of $(U_1(P), \ldots, U_k(P))$, $(V_1(P), \ldots, V_k(P))$ and $(I_1(P), \ldots, I_k(P))$ in T.

Because the inclusion $\mathsf{DNPP}_k(T) \subseteq \mathsf{DNPP}_k(P)$ in Theorem 6.4.2 follows alone from the regularity of P we infer that these tuples are also representable in the proof system P. □

Combining Theorem 6.3.2 and Lemma 6.4.3 we conclude:

Corollary 6.4.4 *Let P be a regular proof system that is closed under substitutions by constants. Then for every $k \geq 2$ the pair $(U_1(P), \ldots, U_k(P))$ is $\leq_s$-complete for $\mathsf{DNPP}_k(P)$.*

For strongly regular proof systems P we can additionally show the $\leq_s$-completeness of the k-tuple $(I_1(P), \ldots, I_k(P))$ for $\mathsf{DNPP}_k(P)$, thereby extending Theorem 4.8.7 to k-tuples:

Theorem 6.4.5 *Let $P \geq EF$ be a strongly regular proof system that is efficiently closed under substitutions by constants. Then for all $k \geq 2$ the tuples $(U_1(P), \ldots, U_k(P))$ and $(I_1(P), \ldots, I_k(P))$ are $\leq_s$-complete for $\mathsf{DNPP}_k(P)$. In particular we have*

$$(U_1(P), \ldots, U_k(P)) \equiv_s (I_1(P), \ldots, I_k(P)) \ .$$

Proof. The $\leq_s$-completeness of $(U_1(P), \ldots, U_k(P))$ was already stated in Corollary 6.4.4.

As by Lemma 6.4.3 also $(I_1(P), \ldots, I_k(P))$ is representable in P it remains to show that $(I_1(P), \ldots, I_k(P))$ is $\leq_s$-hard for $\mathsf{DNPP}_k(P)$. For this let $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets that is representable in P. By Theorem 6.4.2 we know that $(A_1, \ldots, A_k)$ is also representable in the theory T corresponding to P. Let $\varphi_i(x)$ be arithmetic representations of A_i for $i = 1, \ldots, k$ such that

$$T \vdash (\forall x) \bigwedge_{1 \leq i < j \leq k} \neg\varphi_i(x) \vee \neg\varphi_j(x) \ .$$

Because this is a $\forall\Pi^b_1$-formula and P is strongly regular there exists a polynomial time computable function f that on input 1^n produces a P-proof of

$$\| \bigwedge_{1\le i<j\le k} \neg\varphi_i(x) \vee \neg\varphi_j(x)\|^n \ .$$

Further, because by assumption P is efficiently closed under substitutions by constants we can use f to obtain a polynomial time computable function g that on input $\bar{a} \in \{0,1\}^n$ outputs a P-proof of

$$\| \bigwedge_{1\le i<j\le k} \neg\varphi_i(x) \vee \neg\varphi_j(x)\|^n(\bar{p}^x/\bar{a})$$

where the propositional variables $\bar{p}^x$ for x are substituted by the bits of a.

We claim that the $\le_s$-reduction from $(A_1,\dots,A_k)$ to $(I_1(P),\dots,I_k(P))$ is given by

$$a \mapsto ((\|\neg\varphi_i(x)\|^{|a|}(\bar{p}^x/\bar{a}) \mid 1 \le i \le k), g(\bar{a}))$$

where the auxiliary variables of $\|\neg\varphi_i(x)\|^{|a|}$ are all chosen disjoint. Verifying the correctness of the reduction then proceeds as in the proof of Theorem 4.8.3. □

As a corollary we get from Proposition 6.2.2 and Theorem 6.4.5 for the extended Frege system EF:

Corollary 6.4.6 *For every number $k \ge 2$ and every k-tuple $(A_1,\dots,A_k)$ of* NP*-sets we have $(A_1,\dots,A_k) \in \mathsf{DNPP}_k(EF)$ if and only if $(A_1,\dots,A_k) \le_s (U_1(EF),\dots,U_k(EF))$.*

Additionally, we have

$$(U_1(EF),\dots,U_k(EF)) \equiv_s (I_1(EF),\dots,I_k(EF)) \ .$$

The corollary is also true for all extensions $EF + \|\Phi\|$ of the extended Frege systems for polynomial time sets Φ of true Π^b_1-formulas.

6.5 On Complete Disjoint Tuples of NP-Sets

In this section we will study the question whether there exist complete disjoint k-tuples of NP-sets under the reductions $\le_p$ and $\le_s$. We will not be able to answer this question but we will relate it to the previously studied questions whether there exist complete disjoint NP-pairs or optimal propositional proof systems. The following is the main theorem of this section:

Theorem 6.5.1 *The following conditions are equivalent:*

1. *For all numbers $k \geq 2$ there exists a $\leq_s$-complete disjoint k-tuple of* NP*-sets.*
2. *For all numbers $k \geq 2$ there exists a $\leq_p$-complete disjoint k-tuple of* NP*-sets.*
3. *There exists a $\leq_p$-complete disjoint* NP*-pair.*
4. *There exists a number $k \geq 2$ such that there exists a $\leq_p$-complete disjoint k-tuple of* NP*-sets.*
5. *There exists a propositional proof system P such that for all numbers $k \geq 2$ all disjoint k-tuples of* NP*-sets are representable in P.*
6. *There exists a propositional proof system P such that all disjoint* NP*-pairs are representable in P.*
7. *There exists a propositional proof system P and a number $k \geq 2$ such that all disjoint k-tuples of* NP*-sets are representable in P.*

Proof. To show the equivalence of 1 to 7 we will prove the following implications: $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 6 \Rightarrow 1$ and the equivalences $3 \Leftrightarrow 4$, $5 \Leftrightarrow 6$ and $6 \Leftrightarrow 7$.

As the implications $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4$ and $5 \Rightarrow 6 \Rightarrow 7$ are trivial it remains to prove $3 \Rightarrow 6 \Rightarrow 1$, $4 \Rightarrow 3$, $6 \Rightarrow 5$ and $7 \Rightarrow 6$.

To prove the implication $3 \Rightarrow 6$ assume that (A, B) is a $\leq_p$-complete disjoint NP-pair. We choose some representations φ_n and ψ_n for A and B, respectively. Let P be a proof system such that (A, B) is representable in P, and P simulates resolution and is closed under disjunctions. For instance the proof system

$$EF + \{\neg\varphi_n \vee \neg\psi_n \mid n \geq 0\}$$

fulfills these conditions. Because (A, B) is representable in P and $\mathsf{DNPP}(P)$ is closed under $\leq_p$ by Proposition 4.6.1, it follows that all disjoint NP-pairs are representable in the system P.

Next we prove the implication $6 \Rightarrow 1$. Let P be a propositional proof system such that all disjoint NP-pairs are representable in P. We choose a proof system $Q \geq P$ that is closed under conjunctions and substitutions by constants. As Q simulates P also the class $\mathsf{DNPP}(Q)$ contains all disjoint NP-pairs. We claim that for all $k \geq 2$ the pair $(V_1(Q), \ldots, V_k(Q))$ is $\leq_s$-complete for the class of all disjoint k-tuples of NP-sets. To verify the claim let $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets. In particular, for all $1 \leq i <$

$j \leq k$ the pair (A_i, A_j) is a disjoint NP-pair. By assumption all these pairs are representable in Q. However, we might need different representations for the sets A_i to prove the disjointness of all these pairs. For example proving $A_1 \cap A_2 = \emptyset$ and $A_1 \cap A_3 = \emptyset$ might require two different propositional representations for A_1. For this reason we cannot simply reduce $(A_1, \ldots, A_k)$ to $(U_1(Q), \ldots, U_k(Q))$. But we can reduce $(A_1, \ldots, A_k)$ to $(V_1(Q), \ldots, V_k(Q))$ which was designed for this particular purpose.

For $1 \leq i < j \leq k$ let $\varphi_n^{i,j}(\bar{x}, \bar{y}^{i,j})$ and $\varphi_n^{j,i}(\bar{x}, \bar{y}^{j,i})$ be propositional representations of A_i and A_j, respectively, such that all tuples of variables $\bar{y}^{i,j}$ are chosen distinct and

$$Q \vdash_* \neg\varphi_n^{i,j}(\bar{x}, \bar{y}^{i,j}) \vee \neg\varphi_n^{j,i}(\bar{x}, \bar{y}^{j,i}) \ .$$

Because Q is closed under conjunctions we can combine all these proofs to obtain

$$Q \vdash_* \bigwedge_{i=1}^{k} \bigwedge_{j=i+1}^{k} \neg\varphi_n^{i,j}(\bar{x}, \bar{y}^{i,j}) \vee \neg\varphi_n^{j,i}(\bar{x}, \bar{y}^{j,i}) \ . \tag{6.3}$$

The reduction from $(A_1, \ldots, A_k)$ to $(V_1(Q), \ldots, V_k(Q))$ is given by

$$a \mapsto ((\neg\varphi_n^{i,j}(\bar{a}, \bar{y}^{i,j}) \mid 1 \leq i, j \leq k, i \neq j), 1^{p(m)})$$

for some appropriate polynomial p which comes from (6.3) and the closure of Q under substitutions by constants. To prove the correctness of the reduction let a be an element from A_r for some $r \in \{1, \ldots, k\}$. As for all $j \in \{1, \ldots, k\} \setminus \{r\}$ the sequences $\varphi_n^{r,j}$ are representations for A_r all formulas $\varphi_n^{r,j}(\bar{a}, \bar{y}^{r,j})$ are satisfiable. By substituting the bits $\bar{a}$ of a for the variables $\bar{x}$ we get from (6.3) polynomial size Q-proofs of

$$\bigwedge_{i=1}^{k} \bigwedge_{j=i+1}^{k} \neg\varphi_n^{i,j}(\bar{a}, \bar{y}^{i,j}) \vee \neg\varphi_n^{j,i}(\bar{a}, \bar{y}^{j,i}) \ .$$

This shows $((\neg\varphi_n^{i,j}(\bar{a}, \bar{y}^{i,j}) \mid 1 \leq i, j \leq k, i \neq j), 1^{p(m)}) \in V_r(Q)$.

If a is in the complement of $A_1 \cup \ldots \cup A_k$, then none of the formulas $\varphi_n^{i,j}(\bar{a}, \bar{y}^{i,j})$ is satisfiable and hence a is mapped to a tuple from the complement of $V_1(Q) \cup \ldots \cup V_k(Q)$.

We proceed with the proof of the implication $4 \Rightarrow 3$. Assume that the tuple $(A_1, \ldots, A_k)$ is $\leq_p$-complete for all disjoint k-tuples of NP-sets. We claim that (A_1, A_2) is a $\leq_p$-complete disjoint NP-pair. To prove this let (B_1, B_2) be an arbitrary disjoint NP-pair. Without loss of generality we may assume that the complement of $B_1 \cup B_2$ contains at least $k - 2$ distinct

elements $b_3, \ldots, b_k$, because otherwise we can change from (B_1, B_2) to a $\leq_p$-equivalent pair with this property. Since $(A_1, \ldots, A_k)$ is $\leq_p$-complete for all k-tuples there exists a reduction f from $(B_1, B_2, \{b_3\}, \ldots, \{b_k\})$ to $(A_1, \ldots, A_k)$. In particular f is then a reduction from (B_1, B_2) to (A_1, A_2).

Next we prove the implication $6 \Rightarrow 5$. Let P be a proof system such that all disjoint NP-pairs are representable in P. We choose a regular proof system Q that simulates P and is closed under conjunctions, disjunctions and substitutions by constants, for example $Q = EF + \|\mathrm{RFN}(P)\|$ is such a system. Clearly, every disjoint NP-pair is also representable in Q. Going back to the proof of $6 \Rightarrow 1$ we see that condition 6 implies that for all $k \geq 2$ the k-tuple $(V_1(Q), \ldots, V_k(Q))$ is $\leq_s$-complete for the class of all disjoint k-tuples of NP-sets. By Lemma 6.4.3 $(V_1(Q), \ldots, V_k(Q))$ is representable in Q and by Proposition 6.2.2 the class $\mathsf{DNPP}_k(Q)$ is closed under $\leq_s$. Hence for all $k \geq 2$ all disjoint k-tuples of NP-sets are representable in Q.

The last part of the proof is the implication $7 \Rightarrow 6$. For this let P be a proof system and k be a number such that all disjoint k-tuples of NP-sets are representable in P. We choose some proof system Q that simulates P and is closed under conjunctions. As $Q \geq P$ all disjoint k-tuples of NP-sets are representable in Q. To show that also all disjoint NP-pairs are representable in the system Q let (B_1, B_2) be a disjoint NP-pair. As in the proof of $4 \Rightarrow 3$ we stretch (B_1, B_2) to a disjoint k-tuple $(B_1, B_2, \{b_3\}, \ldots, \{b_k\})$ with some elements $b_3, \ldots, b_k \in \overline{B_1 \cup B_2}$. By assumption $(B_1, B_2, \{b_3\}, \ldots, \{b_k\})$ is representable in Q via some representations $\varphi_n^1, \ldots, \varphi_n^k$. Because Q is closed under conjunctions this implies that Q proves the disjointness of B_1 and B_2 with respect to φ_n^1 and φ_n^2, hence (B_1, B_2) is representable in Q. □

We can also characterize the existence of complete disjoint k-tuples of NP-sets by conditions on arithmetic theories, thereby extending the list of characterizations from Theorem 6.5.1 by the items listed in the next theorem:

Theorem 6.5.2 *The following conditions are equivalent:*

1. *For all numbers $k \geq 2$ there exists a $\leq_s$-complete disjoint k-tuple of NP-sets.*

2. *There exists a finitely axiomatized arithmetic theory T such that for all numbers $k \geq 2$ all disjoint k-tuples of NP-sets are representable in T.*

3. *There exists an arithmetic theory T with a polynomial time set of axioms such that for some number $k \geq 2$ all disjoint k-tuples of NP-sets are representable in T.*

Proof. We start with the proof of the implication $1 \Rightarrow 2$. By Theorem 6.5.1 we know already that condition 1 implies the existence of a proof system P in which all disjoint k-tuples of NP-sets are representable. Because by Proposition 3.7.5 P is simulated by the proof system $EF + \|\mathrm{RFN}(P)\|$ all k-tuples are also representable in $EF + \|\mathrm{RFN}(P)\|$. By Theorem 3.6.9 this system is regular and corresponds to the theory $S^1_2 + \mathrm{RFN}(P)$. Therefore all disjoint k-tuples of NP-sets are representable in $S^1_2 + \mathrm{RFN}(P)$ by Theorem 6.4.2. As the theory S^1_2 is finitely axiomatizable (cf. (Kra95)) we have proven condition 2.

As condition 3 obviously is a weakening of condition 2 it remains to prove $3 \Rightarrow 1$. For this let $k \geq 2$ be a natural number and T be an arithmetic theory such that $\mathsf{DNPP}_k(T)$ contains all disjoint k-tuples of NP-sets. Consider the theory $T' = T \cup S^1_2$. As T' is an extension of T all k-tuples are also representable in T'. As in (KP89) we define from the theory T' a propositional proof system P as follows:

$$P(\pi) = \begin{cases} \varphi & \text{if } \pi \text{ is a } T'\text{-proof of } \mathrm{Taut}(\varphi) \\ \top & \text{otherwise.} \end{cases}$$

Because T' has a polynomial time axiomatization this defines indeed a propositional proof system. We claim that all k-tuples are representable in P. To verify this claim let $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets. By hypothesis there exist arithmetic representations $\varphi_1, \ldots, \varphi_k$ of $A_1, \ldots, A_k$ such that

$$T \vdash (\forall x) \bigwedge_{1 \leq i < j \leq k} \neg\varphi_i(x) \vee \neg\varphi_j(x) \ . \tag{6.4}$$

From Lemma 3.6.5 we know that for Π^b_1-formulas ψ we have

$$S^1_2 \vdash (\forall x)\psi(x) \rightarrow (\forall y)\mathrm{Taut}(\|\psi\|^{|y|}) \ .$$

Therefore we get from (6.4)

$$T' \vdash (\forall y)\mathrm{Taut}(\| \bigwedge_{1 \leq i < j \leq k} \neg\varphi_i(x) \vee \neg\varphi_j(x)\|^{|y|}) \ .$$

By the construction of P this implies

$$P \vdash_* \| \bigwedge_{1 \leq i < j \leq k} \neg\varphi_i(x) \vee \neg\varphi_j(x)\|^n \ . \tag{6.5}$$

The translations $\|\varphi_i\|^n$ are propositional representations for the components A_i for $i = 1, \ldots, k$. By the definition of the translations $\|.\|$ we get from (6.5)

$$P \vdash_* \bigwedge_{1 \leq i < j \leq k} \neg\|\varphi_i(x)\|^n \vee \neg\|\varphi_j(x)\|^n \ ,$$

hence $(A_1, \ldots, A_k)$ is representable in P. Therefore, all disjoint k-tuples of NP-sets are representable in P which by Theorem 6.5.1 implies condition 1. □

In Theorem 6.5.1 we stated that the existence of complete disjoint NP-pairs is equivalent to the existence of a propositional proof system P in which every disjoint NP-pair is representable. By definition this condition means that for all disjoint NP-pairs there exists a representation for which the disjointness of the pair is provable with short P-proofs. If we strengthen this condition by requiring that this is possible for all disjoint NP-pairs and all representations we arrive at a condition which is strong enough to characterize the existence of optimal proof systems. This is the contents of the next theorem.

Theorem 6.5.3 *The following conditions are equivalent:*

1. *There exists an optimal propositional proof system.*
2. *There exists a propositional proof system P such that for all $k \geq 2$ the system P proves the disjointness of all disjoint k-tuples of NP-sets with respect to all representations, i.e. for all disjoint k-tuples $(A_1, \ldots, A_k)$ of NP-sets and all representations $\varphi_n^1, \ldots, \varphi_n^k$ of $A_1, \ldots, A_k$ we have $P \vdash_* \bigwedge_{1 \leq i < j \leq k} \neg\varphi_n^i \vee \neg\varphi_n^j$.*
3. *There exists a propositional proof system P that proves the disjointness of all disjoint NP-pairs with respect to all representations, i.e. for all disjoint NP-pairs (A, B) and all representations φ_n of A and ψ_n of B we have $P \vdash_* \neg\varphi_n \vee \neg\psi_n$.*
4. *There exists a propositional proof system P and a number $k \geq 2$ such that P proves the disjointness of all disjoint k-tuples of NP-sets with respect to all representations.*

Proof. To prove the implication $1 \Rightarrow 2$ let P be an optimal proof system. Let further $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets and let φ_n^i be propositional representations of A_i for $i = 1, \ldots, k$. As the sequence of tautologies

$$\bigwedge_{1 \leq i < j \leq k} \neg\varphi_n^i \vee \neg\varphi_n^j$$

can be generated in polynomial time we can define some proof system Q with $Q \vdash_* \bigwedge_{1 \leq i < j \leq k} \neg\varphi_n^i \vee \neg\varphi_n^j$. But because P is optimal we have $Q \leq P$ and therefore also $P \vdash_* \bigwedge_{1 \leq i < j \leq k} \neg\varphi_n^i \vee \neg\varphi_n^j$.

As $2 \Rightarrow 3$ and $3 \Rightarrow 4$ trivially hold it only remains to show $4 \Rightarrow 1$. For this assume that optimal proof systems do not exist. To prove that condition 4 fails let k be a natural number and let P be a proof system. We choose some proof system Q that simulates P and is closed under conjunctions. Let $(A_1, \ldots, A_k)$ be a disjoint k-tuple of NP-sets. Then we know by Corollary 4.5.6 that there exist representations φ_n^1 and φ_n^2 for A_1 and A_2, respectively, such that the DNPP (A_1, A_2) is not representable in Q with respect to φ_n^1 and φ_n^2, i.e. $Q \nvdash_* \neg\varphi_n^1 \vee \neg\varphi_n^2$. We choose arbitrary representations $\varphi_n^3, \ldots, \varphi_n^k$ for $A_3, \ldots, A_k$. As Q is closed under conjunctions Q does not prove the disjointness of $(A_1, \ldots, A_k)$ with respect to $\varphi_n^1, \ldots, \varphi_n^k$ and as $P \leq Q$ this is also true for the system P. Hence condition 4 fails. □

As an immediate corollary to Theorems 6.5.1 and 6.5.3 we get a strengthening of a theorem of Köbler, Messner and Torán (KMT03), stating that the existence of optimal proof systems implies the existence of $\leq_s$-complete disjoint NP-pairs:

Corollary 6.5.4 *If there exist optimal proof systems, then there exist $\leq_s$-complete disjoint k-tuples of NP-sets for all numbers $k \geq 2$.*

Proof. The existence of optimal proof systems implies condition 2 of Theorem 6.5.3. This condition is a strengthening of condition 5 from Theorem 6.5.1 which is equivalent to the existence of $\leq_s$-complete disjoint k-tuples of NP-sets for all $k \geq 2$. □

Bibliography

[AB87] Alon, Noga; Boppana, Ravi B.: The monotone circuit complexity of boolean functions. In: *Combinatorica*, volume 7:pp. 1–22, 1987.

[AB02] Atserias, Albert; Bonet, Maria Luisa: On the automatizability of resolution and related propositional proof systems. In: *Computer Science Logic, 16th International Workshop*, pp. 569–583. 2002.

[ABSRW04] Alekhnovich, Michael; Ben-Sasson, Eli; Razborov, Alexander A.; Wigderson, Avi: Pseudorandom generators in propositional proof complexity. In: *SIAM Journal on Computing*, volume 34(1):pp. 67–88, 2004.

[Ajt94] Ajtai, Miklós: The complexity of the pigeonhole-principle. In: *Combinatorica*, volume 14(4):pp. 417–433, 1994.

[AR01] Alekhnovich, Michael; Razborov, Alexander A.: Resolution is not automatizable unless W[P] is tractable. In: *Proc. 42nd IEEE Foundations of Computer Science*, pp. 210–219. 2001.

[BB93] Bonet, Maria Luisa; Buss, Samuel R.: The deduction rule and linear and near-linear proof simulations. In: *The Journal of Symbolic Logic*, volume 58(2):pp. 688–709, 1993.

[BBP95] Bonet, Maria Luisa; Buss, Samuel R.; Pitassi, Toniann: Are there hard examples for Frege systems? In: Clote, P.; Remmel, J., editors, *Feasible Mathematics II*, pp. 30–56. Birkhäuser, 1995.

[BDG88] Balcázar, José L.; Díaz, Josep; Gabarró, Joaquim: *Structural Complexity I.* Springer-Verlag, Berlin Heidelberg, 1988.

[BDG+04] Bonet, Maria Luisa; Domingo, Carlos; Gavaldà, Ricard; Maciel, Alexis; Pitassi, Toniann: Non-automatizability of bounded-depth Frege proofs. In: *Computational Complexity*, volume 13(1-2):pp. 47–68, 2004.

[Bey04a] Beyersdorff, Olaf: Representable disjoint NP-pairs. In: *Proc. 24th Conference on Foundations of Software Technology and Theoretical Computer Science*, pp. 122–134. 2004.

[Bey04b] Beyersdorff, Olaf: Representable disjoint NP-pairs. Technical Report TR04-082, Electronic Colloquium on Computational Complexity, 2004.

[Bey05a] Beyersdorff, Olaf: Disjoint NP-pairs from propositional proof systems. Technical Report TR05-083, Electronic Colloquium on Computational Complexity, 2005.

[Bey05b] Beyersdorff, Olaf: Tuples of disjoint NP-sets. Technical Report TR05-123, Electronic Colloquium on Computational Complexity, 2005.

[Bey06a] Beyersdorff, Olaf: Disjoint NP-pairs from propositional proof systems. In: *Proc. 3rd Conference on Theory and Applications of Models of Computation*, pp. 236–247. 2006.

[Bey06b] Beyersdorff, Olaf: Tuples of disjoint NP-sets. In: *Proc. 1st International Computer Science Symposium in Russia*, pp. 80–91. 2006.

[BGS75] Baker, Theodore; Gill, John; Solovay, Robert: Relativizations of the P=?NP question. In: *SIAM Journal on Computing*, volume 4:pp. 431–442, 1975.

[Bon93] Bonet, Maria Luisa: Number of symbols in Frege proofs with and without the deduction rule. In: Clote, P.; Krajíček, J., editors, *Arithmetic, Proof Theory and Computational Complexity*, pp. 61–95. Oxford University Press, Oxford, 1993.

[BPR97] Bonet, Maria Luisa; Pitassi, Toniann; Raz, Ran: Lower bounds for cutting planes proofs with small coefficients. In: *The Journal of Symbolic Logic*, volume 62(3):pp. 708–728, 1997.

[BPR00] Bonet, Maria Luisa; Pitassi, Toniann; Raz, Ran: On interpolation and automatization for Frege systems. In: *SIAM Journal on Computing*, volume 29(6):pp. 1939–1967, 2000.

[Bus86] Buss, Samuel R.: *Bounded Arithmetic*. Bibliopolis, Napoli, 1986.

[Bus90] Buss, Samuel R.: Axiomatizations and conservation results for fragments of bounded arithmetic. In: *Logic and Computation, Contemporary Mathematics*, volume 106:pp. 57–84, 1990.

[Bus95] Buss, Samuel R.: Relating the bounded arithmetic and polynomial-time hierarchies. In: *Annals of Pure and Applied Logic*, volume 75:pp. 67–77, 1995.

[Bus98a] Buss, Samuel R.: First order proof theory of arithmetic. In: Buss, Samuel R., editor, *Handbook of Proof Theory*, pp. 79–147. Elsevier, Amsterdam, 1998.

[Bus98b] Buss, Samuel R.: An introduction to proof theory. In: Buss, Samuel R., editor, *Handbook of Proof Theory*, pp. 1–78. Elsevier, Amsterdam, 1998.

[CH99] Cook, Stephen A.; Haken, Amin: An exponential lower bound to the size of monotone real circuits. In: *Journal of Computer and System Sciences*, volume 58:pp. 326–335, 1999.

[Coo71] Cook, Stephen A.: The complexity of theorem proving procedures. In: *Proc. 3rd Annual ACM Symposium on Theory of Computing*, pp. 151–158. 1971.

[Coo75] Cook, Stephen A.: Feasibly constructive proofs of the propositional calculus. In: *Proc. 7th Annual ACM Symposium on Theory of Computing*, pp. 83–97. 1975.

[CR79] Cook, Stephen A.; Reckhow, Robert A.: The relative efficiency of propositional proof systems. In: *The Journal of Symbolic Logic*, volume 44:pp. 36–50, 1979.

[Cra57] Craig, William: Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. In: *The Journal of Symbolic Logic*, volume 22:pp. 269–285, 1957.

[Dow85] Dowd, Martin: Model-theoretic aspects of P≠NP, 1985. Unpublished manuscript.

[DP60] Davis, Martin; Putnam, Hilary: A computing procedure for quantification theory. In: *Journal of the ACM*, volume 7:pp. 210–215, 1960.

[ESY84] Even, Shimon; Selman, Alan L.; Yacobi, Yacov: The complexity of promise problems with applications to public-key cryptography. In: *Information and Control*, volume 61:pp. 159–173, 1984.

[Gen35] Gentzen, Gerhard: Untersuchungen über das logische Schließen. In: *Mathematische Zeitschrift*, volume 39:pp. 68–131, 1935.

[Goe68] Goethe, Johann Wolfgang: *Sämmtliche Werke in sechsunddreißig Bänden*. Verlag der J. G. Cotta'schen Buchhandlung, Stuttgart, 1868.

[Gol05] Goldreich, Oded: On promise problems (a survey in memory of Shimon Even [1935-2004]). Technical Report TR05-018, Electronic Colloquium on Computational Complexity, 2005.

[GS88] Grollmann, Joachim; Selman, Alan L.: Complexity measures for public-key cryptosystems. In: *SIAM Journal on Computing*, volume 17(2):pp. 309–335, 1988.

[GSS05] Glaßer, Christian; Selman, Alan L.; Sengupta, Samik: Reductions between disjoint NP-pairs. In: *Information and Computation*, volume 200(2):pp. 247–267, 2005.

[GSSZ04] Glaßer, Christian; Selman, Alan L.; Sengupta, Samik; Zhang, Liyu: Disjoint NP-pairs. In: *SIAM Journal on Computing*, volume 33(6):pp. 1369–1416, 2004.

[GSZ05] Glaßer, Christian; Selman, Alan L.; Zhang, Liyu: Canonical disjoint NP-pairs of propositional proof systems. In: *Proc. 30th International Symposium on the Mathematical Foundations of Computer Science*, pp. 399–409. 2005.

[HP93] Hájek, Petr; Pudlák, Pavel: *Metamathematics of First-Order Arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin Heidelberg, 1993.

[HS92] Homer, Steven; Selman, Alan L.: Oracles for structural properties: The isomorphism problem and public-key cryptography. In: *Journal of Computer and System Sciences*, volume 44(2):pp. 287–301, 1992.

[Kar72] Karp, Richard M.: Reducibility among combinatorial problems. In: Miller, R. E.; Thatcher, J. W., editors, *Complexity of Computer Computations*, pp. 85–103. Plenum Press, 1972.

[KMT03] Köbler, Johannes; Messner, Jochen; Torán, Jacobo: Optimal proof systems imply complete sets for promise classes. In: *Information and Computation*, volume 184:pp. 71–92, 2003.

[KP89] Krajíček, Jan; Pudlák, Pavel: Propositional proof systems, the consistency of first order theories and the complexity of computations. In: *The Journal of Symbolic Logic*, volume 54:pp. 1963–1079, 1989.

[KP90] Krajíček, Jan; Pudlák, Pavel: Quantified propositional calculi and fragments of bounded arithmetic. In: *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, volume 36:pp. 29–46, 1990.

[KP98] Krajíček, Jan; Pudlák, Pavel: Some consequences of cryptographical conjectures for S_2^1 and EF. In: *Information and Computation*, volume 140(1):pp. 82–94, 1998.

[KPT91] Krajíček, Jan; Pudlák, Pavel; Takeuti, Gaisi: Bounded arithmetic and the polynomial hierarchy. In: *Annals of Pure and Applied Logic*, volume 52:pp. 143–153, 1991.

[Kra95] Krajíček, Jan: *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1995.

[Kra97] Krajíček, Jan: Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic. In: *The Journal of Symbolic Logic*, volume 62(2):pp. 457–486, 1997.

[Kra01a] Krajíček, Jan: On the weak pigeonhole principle. In: *Fundamenta Mathematicae*, volume 170:pp. 123–140, 2001.

[Kra01b] Krajíček, Jan: Tautologies from pseudo-random generators. In: *Bulletin of Symbolic Logic*, volume 7(2):pp. 197–212, 2001.

[Kra04] Krajíček, Jan: Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds. In: *The Journal of Symbolic Logic*, volume 69(1):pp. 265–286, 2004.

[Lad75] Ladner, Richard E.: On the structure of polynomial-time reducibility. In: *Journal of the ACM*, volume 22:pp. 155–171, 1975.

[Lov79] Lovász, László: On the Shannon capacity of graphs. In: *IEEE Trans. Inform. Theory*, volume 25:pp. 1–7, 1979.

[Pap94] Papadimitriou, Christos H.: *Computational Complexity*. Addison-Wesley, 1994.

[PS98] Pudlák, Pavel; Sgall, Jiri: Algebraic models of computation and interpolation for algebraic proof systems. In: Beame, P. W.; Buss, S. R., editors, *Proof Complexity and Feasible Arithmetic*, volume 39 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 279–296. American Mathematical Society, 1998.

[Pud91] Pudlák, Pavel: Ramsey's theorem in bounded arithmetic. In: Börger, E.; et al., editors, *Computer Science Logic '90*, pp. 308–312. Springer, Berlin, 1991.

[Pud97] Pudlák, Pavel: Lower bounds for resolution and cutting planes proofs and monotone computations. In: *The Journal of Symbolic Logic*, volume 62:pp. 981–998, 1997.

[Pud98] Pudlák, Pavel: The lengths of proofs. In: Buss, Samuel R., editor, *Handbook of Proof Theory*, pp. 547–637. Elsevier, Amsterdam, 1998.

[Pud99] Pudlák, Pavel: On the complexity of propositional calculus. In: *Sets and Proofs, Invited papers from Logic Colloquium'97*, pp. 197–218. Cambridge University Press, 1999.

[Pud03] Pudlák, Pavel: On reducibility and symmetry of disjoint NP-pairs. In: *Theoretical Computer Science*, volume 295:pp. 323–339, 2003.

[PW85] Paris, Jeff; Wilkie, Alec J.: Counting problems in bounded arithmetic. In: *Methods in Mathematical Logic, Proc. 6th Latin American Symposium*, pp. 317–340. 1985.

[Raz85] Razborov, Alexander A.: Lower bounds on the monotone complexity of boolean functions. In: *Doklady Akademii Nauk SSSR*, volume 282:pp. 1033–1037, 1985. English translation in: Soviet Math. Doklady, 31, pp. 354–357.

[Raz94] Razborov, Alexander A.: On provably disjoint NP-pairs. Technical Report TR94-006, Electronic Colloquium on Computational Complexity, 1994.

[Ric53] Rice, Henry G.: Classes of recursively enumerable sets and their decision problems. In: *Trans. Am. Math. Soc.*, volume 74:pp. 358–366, 1953.

[Rob65] Robinson, John Alan: A machine-oriented logic based on the resolution principle. In: *Journal of the ACM*, volume 12:pp. 23–41, 1965.

[RR94] Razborov, Alexander A.; Rudich, Steven: Natural proofs. In: *Proc. 26th ACM Symposium on Theory of Computing*, pp. 204–213. 1994.

[RSA78] Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard M.: A method for obtaining digital signatures and public-key cryptosystems. In: *Communications of the ACM*, volume 21(2):pp. 120–126, February 1978.

[SY04] Suzuki, Toshio; Yamakami, Tomoyuki: Resource bounded immunity and simplicity. In: *Proc. 3rd IFIP International Conference on Theoretical Computer Science*, pp. 81–95. 2004.

[Urq95] Urquhart, Alasdair: The complexity of propositional proofs. In: *Bulletin of Symbolic Logic*, volume 1:pp. 425–467, 1995.

[Ver95] Vereshchagin, Nikolai K.: NP-sets are Co-NP-immune relative to a random oracle. In: *Proc. 3rd Israel Symposium on Theory of Computing and Systems*, pp. 40–45. 1995.

[Wra78] Wrathall, Celia: Rudimentary predicates and relative computation. In: *SIAM Journal on Computing*, volume 7(2):pp. 149–209, 1978.

[Zam96] Zambella, Domenico: Notes on polynomially bounded arithmetic. In: *The Journal of Symbolic Logic*, volume 61(3):pp. 942–966, 1996.

ERKLÄRUNG

Ich erkläre hiermit, dass

- ich die vorliegende Dissertationsschrift „Disjoint NP-Pairs and Propositional Proof Systems“ selbständig und ohne unerlaubte Hilfe angefertigt habe,
- ich mich nicht bereits anderwärts um einen Doktorgrad beworben habe oder einen solchen besitze,
- mir die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät II der Humboldt-Universität zu Berlin bekannt ist.

Berlin, den 20. Februar 2006